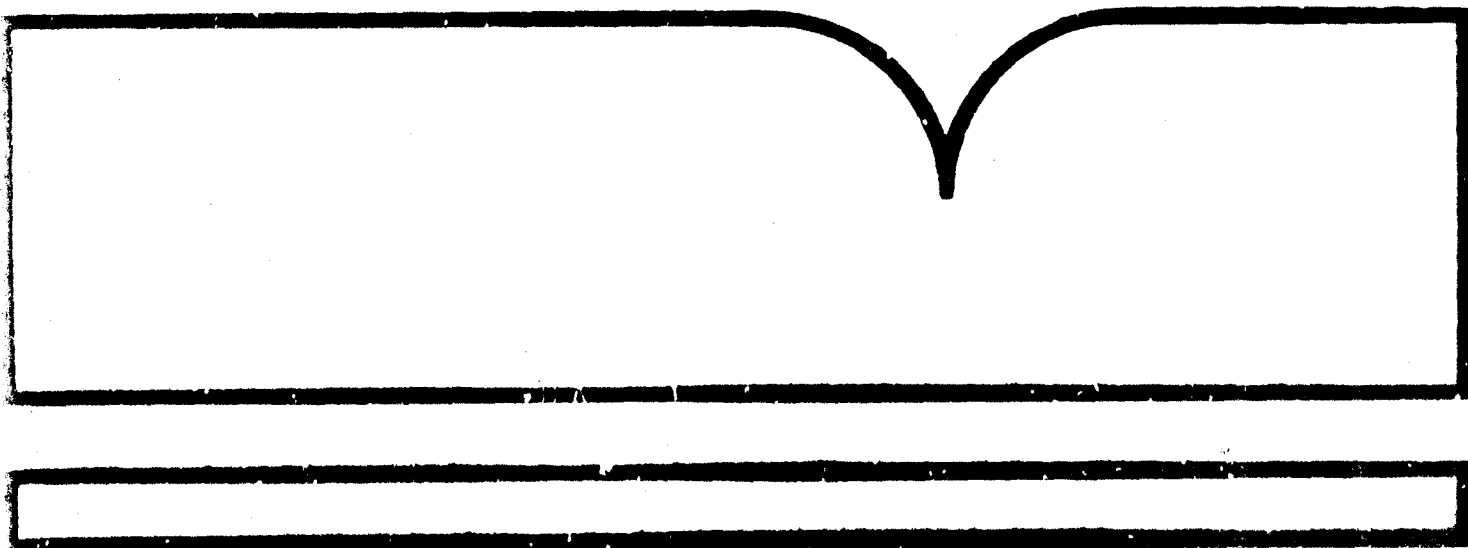


Application of Walsh Functions 1972 Proceedings

R. W. Zeek, et al

29 March 1972



AD-744 650

APPLICATIONS OF WALSH FUNCTIONS

**1972
PROCEEDINGS**

SYMPOSIUM

Sponsors



**Naval Research
Laboratory**



**IEEE - EMC
Group**



**The Catholic University
of America**

REPRODUCED BY
**NATIONAL TECHNICAL
INFORMATION SERVICE**
U. S. DEPARTMENT OF COMMERCE
SPRINGFIELD, VA. 22161

N O T I C E

THIS DOCUMENT HAS BEEN REPRODUCED FROM THE
BEST COPY FURNISHED US BY THE SPONSCRING
AGENCY. ALTHOUGH IT IS RECOGNIZED THAT CER-
TAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RE-
LEASED IN THE INTEREST OF MAKING AVAILABLE
AS MUCH INFORMATION AS POSSIBLE.

APPLICATIONS OF WALSH FUNCTIONS

**1972
PROCEEDINGS**

27,28,29 MARCH

SYMPOSIUM

Held at
**THE CATHOLIC UNIVERSITY
OF AMERICA
Washington, D.C.**

THIRD PROCEEDINGS
**EDITORS: R.W. ZEEK
A.E. SHOWALTER**

PUBLICITY COMMITTEE

ARGENTINA	R. LOPEZ DE ZAVALIA Universidad Nacional de Tucuman
AUSTRALIA	C. YUEN University of Sydney
AUSTRIA	F. PICHLER Hochschule Linz
BELGIUM	M. COGNEAUX Faculté Polytechnique de Mons
CANADA	I. A. DAVIDSON University of Waterloo
FRANCE	C. HUNT Universite de Nice
GERMANY	C. BÖSSWETTER Battelle-Institut
GREAT BRITAIN	J. E. GIBBS National Physical Laboratory
INDIA	S. S. R. MURTHY Regional Engineering College of Andhra Pradesh
ISRAEL	G. LIEBERMAN Israel Aircraft Industries
ITALY	E. BRIGANTI Radiotelevisione Italiana
JAPAN	C. WATARI Tōhoku University
NETHERLANDS	J. L. BORDEWIJK Technological University Delft
SWEDEN	B. OLSEN University of Uppsala
SWITZERLAND	F. BORGNIS Eth. Zuerich
UNITED STATES OF AMERICA	H. SCHLICKE Allen-Bradley Company

PAPERS COMMITTEE

C. A. BASS, Naval Electronics Laboratory Center
J. PEARL, University of California at L.A.
H. L. PETERSON, Naval Research Laboratory

SESSION ORGANIZERS

H. C. ANDREWS, University of Southern California
J. W. BAYLESS, Defense Communications Agency
D. S. GANN, Johns Hopkins University
P. C. JAIN, Stanford Research Institute
J. S. LEE, Catholic University of America
F. R. OHNSORG, Honeywell Inc.
E. F. VANDIVERE, Telcom Inc.

STEERING COMMITTEE

H. F. HARMUTH, Catholic University of America
J. D. LEE, ITT Electro-Physics Laboratories
R. MEISTER, Catholic University of America
L. W. THOMAS, Thomas Engineering Co. (IEEE representative)
J. L. WALSH, University of Maryland
L. B. WETZEL, Naval Research Laboratory
R. W. ZEEK, Naval Research Laboratory

CONFERENCE COORDINATOR

A. E. SHOWALTER, Naval Research Laboratory

INDUSTRIAL PATRONS

Allen-Bradley Co., Milwaukee, Wisconsin
Amaco Production Co., Tulsa, Oklahoma
Computer Signal Processors, Inc., Burlington, Mass.
Grumman Aerospace Corp., Bethpage, N.Y.
Honeywell, Inc., St. Paul, Minn.
Interstate Electronics Corp., Anaheim, Calif.
ITT Electro-Physics Laboratories, Inc., Columbia, Md.
The Johns Hopkins University, APL, Silver Spring, Md.
Sperry Rand, Univac Defense Systems Div., St. Paul, Minn.

SESSION CHAIRMEN

P. C. JAIN, Stanford Research Institute, Arlington, Va.
H. ÜBERALL, Catholic University of America, Washington, D.C.
J. J. FRIEND, Bell Telephone Laboratories, Holmdel, N.J.
J. W. CARL, Aerospace Medical Research Laboratory, Dayton, Ohio
G. S. ROBINSON, Comsat Laboratories, Clarksburg, Md.
J. H. R. ROSENBLOOM, AF Office of Scientific Research, Arlington, Va.
G. F. SANDY, The Mitre Corp., McLean, Va.
N. AHMED, Kansas State University, Manhattan, Kan.
R. D. GITLIN, Bell Telephone Laboratories, Holmdel, N.J.

WELCOMING REMARKS

by

Dr. GENE G. MANNELLA
Dean of the School of Engineering and Architecture
The Catholic University of America

* * *

Copies of the 1970, 1971, and 1972 Proceedings of Walsh Functions Symposia
can be obtained from:

National Technical Information Service
U.S. Department of Commerce
Springfield, Virginia 22151

Order these proceedings by title and identifying AD number. Payment should
accompany the order; however, if billing is requested, a service charge of 50¢
per title will be made.

1970 Proceedings, "Applications of Walsh Functions," AD-707 431
\$3.00 per copy.

1971 Proceedings, "Applications of Walsh Functions," AD-727 000
\$3.00 per copy.

1972 Proceedings, "Applications of Walsh Functions," AD-744 650
\$6.00 per copy.

* * *

CONTENTS

	Page
Welcoming Remarks Dr. G. G. Mannella	xi
WALSH FUNCTIONS SYMPOSIUM	
The Wonderful World of Walsh Functions R. B. Lackey	2
Walsh Functions and Hadamard Transform N. Ahmed and K. R. Rao	8
Walsh Functions in Image Processing and Two-Dimensional Filtering W. K. Pratt	14
All you Always Wanted to Know About Electromagnetic Walsh Waves H. F. Harmuth	23
Walsh Functions in Grille Spectroscopy A. M. Despain and G. A. Vanasse	30
Hadamard Transform Scintillation Counter L. M. Soroko	36
Multiplex Target L. M. Soroko	42
Information Properties of Particle Telescope of Multiplex Scintillation Counters L. M. Soroko	45
Walsh Spectroscopy of Rayleigh Waves Caused by Underground Detonations M. B��th and S. Burman	48
Walsh Domain Processing of Marine Seismic Data C. Chen	64
A Digital Instrument for the Inverse Walsh Transform W. O. Brown and A. R. Elliott	68
Group Multiplexing by Concatenation of Non-Linear Code Division Systems R. A. Barrett and J. A. Gordon	73
Results of Multiplexing Experiments Using Walsh Functions D. I. Durst	82
A Walsh-Function Power-Cable Monitoring System F. Furrer, A. Shah, and M. Maurer	89

Some Considerations in Sequence Multiplexing Systems C. Lin and S. C. Gupta	94
Comparison of Methods for Multiplexing Digital Signals Using Sequence Techniques H. Hübner	100
An Adaptive Digital Voice Multiplexer Using Walsh Functions H. E. Jones	110
Walsh Function Generator for Laboratory Use R. J. Lopez de Zavalía and S. M. Moro ..	114
A New Walsh Generator and Estimation of the Total Orthogonality Error of Generators R. J. Lopez de Zavalía	122
An Orthogonal Transform Approach to the Description of Biological and Medical Systems F. J. Seif and D. C. Schuman	128
A Quantized Variable Approach to Description of Biological and Medical Systems D. S. Gar, F. J. Seif, and J. D. Schoeffler	134
Modelling the Compound Action Potential of the Nerve C. Bösswetter	142
An Application of Walsh Functions to the Monitoring of Electrocardiograph Signals P. J. Milne, N. Ahmed, R. R. Gallagher, and S. G. Harris	149
Heart Rate Representation Using Walsh Functions C. W. Thomas and A. J. Welch	154
Walsh Power Spectra of Human Electroencephalograms W. C. Yeo and J. R. Smith	159
Speech Processing with Walsh Functions H. Gethöffer	163
Word Recognition by Means of Walsh Transforms M. T. Clark, J. E. Swanson, and J. A. Sanders	169
A Programmable Walsh Function Generator and Its Use in a High Speed Inverse Transform Apparatus A. R. Elliott and S. Mikhail	173
Computation of the Fast Hadamard Transform Y. Y. Shum and A. R. Elliott	177
A Parallel Array Hardware Implementation of the Fast Hadamard and Walsh Transforms A. R. Elliott and Y. Y. Shum	181
Speech Analysis and Synthesis Using the Hadamard Transform Y. Y. Shum and A. R. Elliott	184

Digital Simulation of a LCS Resonant Filter for Walsh Functions J. P. Golden	186
Resonant Sequency Filters in the Z-Domain H. T. Nagle	193
Frequency Domain Design of Sequency Filters A. E. Kahveci and E. L. Hall	198
Waveform Analysis of Image Signals by Orthogonal Transformation K. Shibata	210
Walsh Functions for Image Enhancement R. L. Richardson, F. R. Reich, and B. P. Hildebrand	216
A Wired-In Resistor Circuit Realization of the Two-Dimensional Hadamard- Transformation of Broadband Television Signals U. Kraus	224
Slant Transforms for Image Coding W. K. Pratt, L. R. Welch, and W. Chen	229
Modified Transforms in Imagery Analysis G. G. Murray	235
Quantization Noise Considerations in Walsh Transform Image Processing G. S. Robinson	240
Spatial Subsampling for the Transform Coding of Images C. Reader	248
Generalized Sampling Interpretation of Hadamard and Haar Transforms H. H. Schreiber	253
Walsh Functions and the Sampling Principle M. Maqusi	261
Sampling Expansions in Discrete and Finite Walsh-Fourier Analysis C. T. L. Dinh, P. Le and R. Goulet	265
Fast "In Place" Computation of the Discrete Walsh Transform in Sequency Order G. Berauer	272
Efficient Computation of the Walsh-Hadamard Transform Spectral Modes N. Ahmed, A. L. Abdussattar and K. R. Rao	276
Analysis of the Logical Walsh Transform for $N=16$ W. A. Parkyn, Jr. and G. E. Cash	281
Approximation Errors of a Walsh Series C. K. Yuen	289

The Hilbert Schmidt Decorrelation Measure for Fourier and Walsh Transforms J. Pearl	293
Walsh Series to Fourier Series Conversion K. H. Siemens and R. Kitai	295
Discrete Walsh and Fourier Power Spectra G. S. Robinson	298
Mutual Mapping of Generalized Convolution Systems H. Gethöffer	310
A Class of Efficient Convolution Algorithms W. F. Davis	318
Linear Minimum Mean-Square Error Codes G. R. Redinbo	330
On Ordering of a Class of Generalized Walsh Functions S. Chang and T. Joseph	337
Positive Real Functions from Othogonal Functions N. K. Bose	344
Inequivalent Sets of 2^n Two Level Orthogonal Functions G. J. Simmons	348
A New Method for Representing Walsh Functions I. Ross and J. Kelly	359
On Walsh Describing Functions C. T. L. Dinh, R. Goulet, and N. VanHoutte	362
Applications of Walsh Functions to Number Theory R. Binkin and E. L. Hall	369
On Classification of Binary Sequences S. C. Kak	379
On Matrices with Walsh Functions as the Eigenvectors S. C. Kak	384
Approximation by Walsh Polynomials and the Concept of a Derivative P. L. Butzer and H. J. Wagner	388
A New Eye on Walsh Functions C. Cardot	393
Sequency Union H. F. Harmuth	401

INDEX TO AUTHORS

Author	Page	Author	Page
Abdussattar, A. L.	276	Goulet, R.	265, 362
Ahmed, N.	8, 149, 276	Gupta, S. C.	94
Barrett, R. A.	73	Hall, E. L.	198, 369
B��th, M.	48	Harmuth, H. F.	23, 401
Berauer, G.	272	Harris, S. G.	149
Binkin, R.	369	Hildebrand, B. P.	216
Bose, N. K.	344	H��bner, H.	100
B��sswetter, C.	142	Jones, H. E.	110
Brown, W. O.	68	Joseph, T.	337
Burman, S.	48	Kahveci, A. E.	198
Butzer, P. L.	388	Kak, S. C.	379, 384
Cardot, C.	393	Kelly, J. J.	359
Cash, G. E.	281	Kitai, R.	295
Chang, S.	337	Kraus, U.	224
Chen, C.	64	Lackey, R. B.	2
Chen, W.	229	Le, P.	265
Clark, M. T.	169	Lin, C.	94
Davis, W. F.	318	Maqusi, M.	261
Despain, A. M.	30	Maurer, M.	89
de Zavalia, R. L.	114	Mikhail, S.	173
Dinh, C. T. L.	265, 362	Milne, P. J.	149
Durst, D. I.	82	Moro, S. M.	114
Elliott, A. R.	68, 173, 177, 181, 184	Murray, G. G.	235
Furrer, F.	89	Nagle, H. T.	193
Gallagher, R. R.	149	Nawrath, R.	122
Gann, D. S.	134, 128	Parkyn, W. A.	281
Geth��ffer, H.	163, 310	Pearl, J.	293
Golden, J. P.	186	Pratt, W. K.	14, 229
Gordon, J. A.	73		

Author	Page	Author	Page
Rao, K. R.	8, 276	Siemens, K. H.	295
Reader, C.	248	Simmons, G. J.	348
Redinbo, G.	330	Smith, J. R.	159
Reich, F. R.	216	Soroko, L. M.	36, 42, 45
Richardson, R. L.	216	Swanson, J. E.	169
Robinson, G. S.	240, 298	Thomas, C. W.	154
Ross, I.	359	Vanasse, G. A.	30
Sanders, J. A.	169	VanHoutte, N.	362
Schoeffler, J. D.	134	Wagner, H. J.	388
Schreiber, H. H.	253	Welch, A. J.	154
Seif, F. J.	134, 128	Welch, L. R.	229
Shah, A.	89	Yeo, W. C.	159
Shibata, K.	210	Yuen, C. K.	289
Shum, Y. Y.	177, 181, 184		

WELCOMING REMARKS

DR. GENE G. MANNELLA
*Dean of the School of Engineering and Architecture
The Catholic University of America*

It is a gratifying task for me to extend to you on behalf of the Catholic University of America an official welcome to our campus. It is a pleasure for us to host the 1972 Symposium on the Applications of Walsh Functions, and we join with our co-sponsors, the Naval Research Laboratory and the Electromagnetic Compatibility Group of the IEEE, in wishing you a successful meeting.

The program for this year continues the strong international flavor that has come to be a hallmark of this symposium. A total of 64 papers from 11 countries will be presented in the next three days, giving this meeting the broadest representation it has had to date. To our colleagues who have journeyed here from other countries we extend a special welcome and the hope that your visit will be most productive.

Based on the pre-registration and the pattern of symposium attendance established in the past several years, we anticipate a total exceeding 200 persons will attend this symposium. In keeping with the practice of the previous Walsh Functions symposia, the proceedings will again be published and distributed at a nominal charge through the National Technical Information Service, although all registrants will receive a free copy as soon as they are available. This will enable the broadest distribution of the material presented here, so that our colleagues throughout the profession can avail themselves of advances in the applications of Walsh Functions as reported here.

It is clear from the statement of the Symposium objectives published in this year's program that especial interest will again be placed on the application of Walsh and other nonsinusoidal functions. This is certainly in keeping with the mood of today's society which is oriented toward the use of the great store of knowledge built up in the past several decades in finding the solution to many problems which cause us concern. This is not in any way to diminish the need for, or importance of, increasing the body of knowledge in a given field, but only to illustrate that our first duty as engineers and scientists lies in explicitly applying our capability to the ultimate benefit of society.

There is something particularly gratifying in successfully applying the abstract to specific problems. Perhaps in the next three days we will see several exciting examples of this. To that end I wish you well in your symposium and hope that it will be as productive as those that have preceded it.

APPLICATIONS OF WALSH FUNCTIONS

SYMPOSIUM

27,28,29 MARCH

THE WONDERFUL WORLD OF WALSH FUNCTIONS *

Robert B. Lackey

Electrical Engineering Department

The Ohio State University
Columbus, Ohio 43210

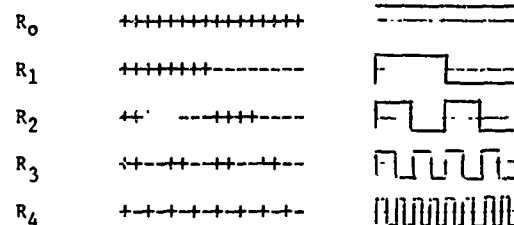
As I pondered the submission of an abstract for this paper, the first title that occurred to me was "Everything you always wanted to know... etc." Aside from the fact that everyone is using that title these days, I am a person who likes to come to the point. I chose instead, "The Wonderful World of Walsh Functions." I will now proceed to answer my own question in as painless a way as I can.

The original paper, by J.L. Walsh, perhaps surprisingly, is not recent; it appeared in print in 1923. As is often the case, with a few notable exceptions, not much was done with these functions, or to them, or about them until 1968. It was then that Dr. H.F. Harmuth's article appeared in the Spectrum of the IEEE, and inspired a tremendous amount of work in the area. It was also at this time that my own personal intimacy with Walsh functions began. The second thing I did after reading the article was to order a copy of Dr. Harmuth's book.[1] I reasoned that if the article were good, the book should be even better, and I was right. (Not always true when they make a movie from a book.) I have been using the book as a text for a course in signal processing with a digital computer. It was during the time that I was waiting for the book to arrive that I began the self-education process which I have found ever so valuable to an ultimate understanding. It is of course my hope that this paper will similarly motivate at least one person to explore the area of Walsh functions more deeply.

Before I tell you how to generate the happy family of Walsh functions, I want to tell you some of their characteristics. Generally speaking, Walsh functions are defined on the unit interval, 0 to 1, and in this interval, they take on the values ± 1 . They flip back and forth between plus and minus one in a fairly regular and highly predictable way. The independent variable on this interval is θ , a normalized kind of time. $\theta = t/T$ where t is ordinary time and T might be called a sampling interval. Walsh functions have some characteristics which make them interesting to mathematicians as well as to engineers. I will just mention these in passing. Walsh functions are orthogonal, normalized, and complete. Orthogonal means that if you multiply

any two functions together and integrate over the interval (add up the values) the answer is zero, unless the two functions are the same function. Normalized means that if the two functions are one and the same, the integral of their product is unity. Complete is a nice property, not worth discussing further here.

As luck would have it, the best way to define the Walsh functions is in terms of another set of functions called Rademacher functions. (This is my own personal favorite definition; others are equally valid). I will define Rademacher functions first. The first step in defining these functions is to take the unit interval and imagine it divided up into 2^n sub-intervals. Did you do it? Okay. Next, a plus sign in a subdivision means +1 and a minus sign means -1. Now I shall list the first few Rademacher functions using this notation with 16 sub-intervals. Let's see if a pattern emerges. (A sketch of each function is shown beside the function to give a picture along with the +, - notation)



Note the following things: the subscript on R is an integer, and the number of complete square-wave cycles the function makes in the interval is $2^{(i-1)}$, when the subscript is i . The exception is R_0 , which is +1 over the whole interval. Easy? You bet.

Now the next step is to define for any who may not know it, the Gray code. If I count in binary from 0 to 7, I can make a table like the one shown below.

Decimal number	Binary	Gray Code
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

*This work was supported in part by Grant GN-534.1 from the Office of Science Information Service, National Science Foundation to the Computer and Information Science Research Center, Ohio State University, Columbus, Ohio.

The Gray code has two interesting properties. (1) As you progress from 0 to 7, only one digit at a time changes. (2) It is helpful in defining Walsh functions. It is particularly easy to convert a number from binary to Gray code. If the binary number is $b_n b_{n-1} b_{n-2} \dots b_1$ then the Gray code is $g_n g_{n-1} g_{n-2} \dots g_1$ and the g 's are found from the b 's by the following rules:

$$\begin{aligned} g_n &= b_n \\ g_{n-1} &= b_n \oplus b_{n-1} \\ g_{n-2} &= b_{n-1} \oplus b_{n-2} \\ &\vdots \\ g_1 &= b_2 \oplus b_1 \end{aligned}$$

The symbol \oplus means addition modulo 2.

Now, at last, we're ready to generate some Walsh functions. A Walsh function is characterized by two parameters. j is the sequency (more about that later) and θ is the independent variable. To find $wal(j, \theta)$

1. Write j in binary
2. Convert j to Gray code
3. Multiply together all Rademacher functions whose subscripts correspond to the position of the 1-bits in the Gray code number. They line up as shown below:

Code word ... $g_4 g_3 g_2 g_1$

Rademacher ... $R_4 R_3 R_2 R_1$

I will do an example, then show you a list of Walsh functions. To find $wal(13, \theta)$

1. 13 = 1101, in binary
2. Gray code = 1011
3. $R_4 \cdot R_2 \cdot R_1$

R_4 + + + + + + + +
 R_2 + + + + - - - -
 R_1 + + + + + + - -

$wal(13, \theta)$ + + - - + + + + - -

So there you have it. A relatively simple way to define Walsh functions. The table below shows the first sixteen of them. I will explain the names on the right later.

$wal(0, \theta)$	+++++	$wal(0, \theta)$
$wal(1, \theta)$	+++++-----	$sal(1, \theta)$
$wal(2, \theta)$	+++-----+++	$cal(1, \theta)$
$wal(3, \theta)$	+++-----+++	$sal(2, \theta)$
$wal(4, \theta)$	++-----++	$cal(2, \theta)$

$wal(5, \theta)$	++---++-----	$sal(3, \theta)$
$wal(6, \theta)$	++---++-----	$cal(3, \theta)$
$wal(7, \theta)$	++---++-----	$sal(4, \theta)$
$wal(8, \theta)$	++---++-----	$cal(4, \theta)$
$wal(9, \theta)$	++---++-----	$sal(5, \theta)$
$wal(10, \theta)$	++---++-----	$cal(5, \theta)$
$wal(11, \theta)$	++---++-----	$sal(6, \theta)$
$wal(12, \theta)$	++---++-----	$cal(6, \theta)$
$wal(13, \theta)$	++---++-----	$sal(7, \theta)$
$wal(14, \theta)$	++---++-----	$cal(7, \theta)$
$wal(15, \theta)$	++---++-----	$sal(8, \theta)$

0 1/2 1 θ axis

Now let's see what some other interesting properties are. Each time one of the functions changes from + to - or - to +, it is an occurrence called a zero-crossing, a most descriptive name. Also, there is a zero-crossing at the origin ($\theta = 0$) for each of those functions which has odd symmetry about $\theta = 1/2$. (they also end with a -). This is true for every other one in the table. If we count the number of zero crossings in each of the previous functions, we find the following numbers: 0, 2, 2, 4, 4, 6, 6, etc. with the last one having 16. Remember the word I used a bit earlier -sequency? Well, sequency is defined as one half the average number of zero-crossings in the interval. With this in mind, we can list the sequencies of the first sixteen Walsh functions. They are, from top to bottom, 0, 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8. In other words, there are two functions, one with odd symmetry and one even, at each sequency from 0 to 8, except for the first and last. In order to point out the similarity between these functions and the also orthonormal set of sines and cosines, the appropriate have been given the names "cal" and "sal" as follows:

$$wal(2n-1, \theta) = sal(n, \theta)$$

$$wal(2n, \theta) = cal(n, \theta)$$

except for 0 sequency where all we have is $wal(0, \theta)$. The index n for sal and cal is their sequency. There are the names listed on the right in the previous table.

Now that we can define Walsh functions at will, and put the appropriate label (cal or sal) on them, as well as find their sequency, we are ready to start working with them. This section might be called "What are they good for?"

Using Walsh functions to represent signals is possibly the most important application today. This application is exactly analogous to the use of sines and cosines to represent a function in

terms of what we all learned as Fourier Series. The difference now is that the function is represented as a series of Walsh functions in the "sequency domain." I would like to digress for just a moment. All the masses of literature that have been released to the scientific community since the famous Cooley-Tukey algorithm appeared in print have referred to the digital Fourier Transform. They call it a discrete version of a continuous process. I prefer to think of it as what it really is: a Fourier Series. If you think this way, then the Fourier Transform becomes a special case, using an infinitely fast sampler. Now let's get back to the subject and talk of a Walsh series for a function. A good way to represent the formulation of a Walsh series is by a matrix equation:

$$[W] \cdot [X] = [A]$$

where $[W]$ is the matrix of Walsh functions, $[X]$ is a column vector of sampled values of the original function, and $[A]$ a column vector of Walsh coefficients.

The Walsh matrix, because of the nature of the functions is simply an array of plus and minus ones. With the functions as defined here, the preceding table is the Walsh matrix for $N = 16$. Now you know why the so-called "Walsh Transform" is so fast. All you do to generate the Walsh coefficients is add and subtract, and if you do it according to the Cooley-Tukey algorithm, it's really fast. Another nice thing about this representation of Walsh functions is that the original signal can be regenerated by multiplying the column vector of Walsh coefficients by the same Walsh matrix, and dividing by N . In other words, the Walsh Matrix is its own inverse, within a constant multiplier N . In terms of a matrix equation,

$$[X] = \frac{1}{N} \cdot [W] \cdot [A] = \frac{1}{N} \cdot [W] \cdot [W] \cdot [X]$$

$$\text{and} \quad [W] \cdot [W] \cdot \frac{1}{N} = [I]$$

where $[I]$ is an $N \times N$ identity matrix. Let's do a simple example, for $N = 8$. Let the sample values be x_0 through x_7 , and the Walsh series coefficients be given by $a_c(n)$ or $a_s(n)$. (Except for the coefficient of $w(0,0)$ which is a (0) .) Here the subscript c or s refers to \cos or \sin .

$a(0)$	++++++	x_0
$a_s(1)$	++++--	x_1
$a_c(1)$	+-+--+	x_2
$a_s(2)$	+-+--+	x_3
$a_c(2)$	+-+--+	x_4
$a_s(3)$	+-+--+	x_5
$a_c(3)$	+-+--+	x_6
$a_s(4)$	+-+--+	x_7

Let the x_i 's take on the following values: 0,0,1,1,0,0,1,1. This is a square wave with sequency 2 and some d.c. A series of additions and subtractions gives the following results:

$a(0) = 4$	$a_c(2) = 0$
$a_s(1) = 0$	$a_s(3) = 0$
$a_c(1) = 0$	$a_c(3) = 0$
$a_s(2) = -4$	$a_s(4) = 0$

$a(0)$ represents the d.c., while $a_s(2)$ is due to the component at sequency 2. Now let's try an interesting thing. Shift the phase of the input signal, by letting the x 's be (in order) 0 1 1 0 0 1 1 0. Now, computation of the Walsh coefficients gives

$a(0) = 4$	$a_c(2) = -4$
$a_s(1) = 0$	$a_s(3) = 0$
$a_c(1) = 0$	$a_c(3) = 0$
$a_s(2) = 0$	$a_s(4) = 0$

This example illustrates that the Walsh transform is sensitive to the phase of the input signal as well as to the sequency. This is not true for the exponential form of Fourier Series, although it is for the sine-cosine form, as might be expected.

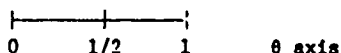
If we sum the squares of the terms at the various sequencies, we have the equivalent of a power spectrum, called appropriately enough, the Walsh spectrum. Note that the spectra for the previous two examples are the same, as they should be; d.c. term and a term at sequency 2.

Let me end this series of ideas with a simple statement. You can mull it over on your own time. A time-limited square wave function (such as one obtains at the output of a sample-and-hold device) can be exactly represented by a finite number of Walsh functions. You can't say that about sines and cosines!

Another interesting set of orthogonal, complete functions, which may be normalized if you desire, is the set of Haar Functions. I will now list enough Haar Functions to allow you to generate as many more as you like. As with the Walsh Functions, the interval is divided into $1/2^n$ spaces. For this list, use sixteen divisions. Capital H with superscripts and subscripts will denote the Haar Functions. The plus sign again means +1, the minus sign means -1, and now the number 0 means exactly that - zero.

Note the following characteristics of the Haar Functions: for the subset H_k , there are $n = 2^{k-1}$ different functions in the interval, and $1 \leq k \leq n$.

H ₁ ⁰	+++++
H ₁ ¹	+++++-----
H ₁ ²	+++-----00000000
H ₂ ²	00000000+++-----
H ₁ ³	+-000000000000
H ₂ ³	0000+-00000000
H ₃ ³	00000000+-0000
H ₄ ³	000000000000+-
H ₁ ⁴	+00000000000000
H ₂ ⁴	00+-000000000000
H ₃ ⁴	0000+-0000000000
H ₄ ⁴	000000+-00000000
H ₅ ⁴	00000000+-000000
H ₆ ⁴	0000000000+-0000
H ₇ ⁴	000000000000+-00
H ₈ ⁴	00000000000000+-



Also note the interesting relationship to the Rademacher Functions.

$$\sum_{k=1}^{2^{j-1}} H_k^j = R_j$$

With this in mind, it is reasonable to talk of a "sequency" for Haar Functions, although a more important parameter is delay, which, for Haar Functions, is analagous to the phase of sines and cosines. For the Haar Functions, the delay is given by $k - 1$. To normalize the Haar Functions of order j , each should be multiplied by $2^{(j-1)/2}$. An application will be described later.

For the remainder of this paper, I shall point out some of the many actual as well as potential applications of Walsh Functions, and other square-wave functions. Perhaps I will be asking more questions than I answer.

1. Two-dimensional image transformation.

Two very active persons in this area are Andrews and Pratt.[2] They were certainly among the first to point out the advantages of using Walsh Functions for image transformations rather than sines and cosines. I thought it might be fun to examine several simple cases in order to see exactly what a Walsh Transform does to an image. As I define the two-dimensional Walsh Transform, it is a matrix operation given by

$$[W][X][W]$$

where $[W]$ is the Walsh matrix and $[X]$ is an image (actually an $n \times n$ portion of an image).

First, consider an image containing information at sequency 1 in the horizontal direction and sequency 0 in the vertical direction, as shown below.

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Image

The transformation is shown below.

$$\begin{bmatrix} + & + & + & + \\ + & + & - & - \\ + & - & - & + \\ + & - & + & - \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} + & + & + & + \\ + & + & - & - \\ + & - & - & + \\ + & - & + & - \end{bmatrix} = \begin{bmatrix} 8 & 0 & -8 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The 8 in the upper left corner is the d.c. or average term and the -8 is due to the sequency 1 component. Note that the -8 lies in the first row, third column, indicating sequency 1 in the horizontal direction.

Next consider an image with 0 sequency in the horizontal direction and sequency 2 in the vertical direction.

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Image

The transformation is shown below.

$$\begin{bmatrix} + & + & + & + \\ + & + & - & - \\ + & - & - & + \\ + & - & + & - \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} + & + & + & + \\ + & + & - & - \\ + & - & - & + \\ + & - & + & - \end{bmatrix} = \begin{bmatrix} 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -8 & 0 & 0 & 0 \end{bmatrix}$$

Note that the d.c. term is in the same place, while the sequency 2 term in the vertical direction is indicated by the -8 in the fourth row, first column.

Finally, using the notation that the Transform array is $[T]$, the original image can be recovered by exactly the same matrix operation as the one which generated it, with a factor of $1/N^2$. This is shown in equation form below.

$$[X] = \frac{1}{N^2} \cdot [W] \cdot [T] \cdot [W]$$

$$[X] = \frac{1}{N} \cdot [W] \cdot [W] \cdot [X] \cdot \frac{1}{N} \cdot [W] \cdot [W]$$

$$[X] = [I] \cdot [X] \cdot [I] = [X]$$

$[I]$ is the identity matrix.

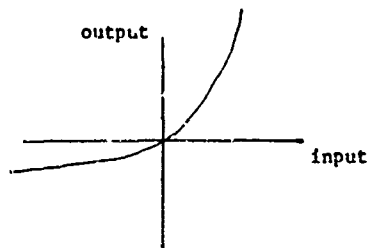
2. Sequency filtering.

It is possible to do filtering based upon sequency rather than frequency in two different ways: the first way utilizes sample and hold amplifiers, switches, integrators, multipliers, counters, etc. Any type of filter can be

synthesized; lowpass, bandpass, highpass, etc. [3] The other approach is to use a digital computer, perform the transformation to a Walsh series, then retain for outputting only those sequence terms which are to be passed by the filter. The feature that makes this approach feasible is the speed with which the transform can be computed. There will of course be a time delay in both cases. Note that a sequence filter can differentiate between s_1 and s_2 . An RLC filter has no such capability.

3. Non-linear system analysis.

A non-linear system is often described by a transfer characteristic as shown below.



If analysis by sine-cosine input waves is performed, the output will be distorted sine wave, expressible as a fundamental and harmonics of a Fourier Series. If, however, a square wave is the input, the output will also be a square wave at exactly the same frequency with d.c. bias. Now all that is left is to make something out of that.

4. Spectroscopy.

The two-beam interferometer spectrometer works by first forming an interferogram, then by means of an inverse Fourier Transform, reconstructing the spectrum. Interest in this technique is currently high because of the decreasing cost and increasing speed of performing this transformation digitally. Interestingly enough, the processing can be done by using a mask with transparent and opaque portions encoded so as to perform a Walsh (Hadamard) transform. The result seems to be increased speed with no loss of resolution.[4]

5. Multiplexing.

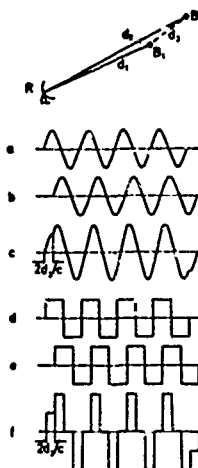
The multiplication theorems of sine and cosine, applicable to amplitude modulation, yield upper and lower sidebands. This is not true in the case of Walsh Functions. The multiplication of $wal(j, \theta)$ by $wal(k, \theta)$ yields $wal[(j \oplus k), \theta]$ where \oplus stands for bit-by-bit addition modulo 2. So you might say a Walsh amplitude modulation has a single, sometimes upper, sometimes lower side-band. This feature leads to Walsh function multiplexing which is most interesting.[5]

6. Radiation of Walsh waves.[1]

Perhaps the most important reason to be considering the radiation of Walsh waves (electromagnetic or acoustic) is this: there is a difference between a Walsh wave at frequency s , and another Walsh wave transmitted at frequency s' but Doppler shifted to frequency s . The application of this feature should be apparent.

7. Resolution of point targets.

If two closely spaced point targets are illuminated with a sine wave, the received signal will have the appearance of the curves below due to small differences in path length.



The composite waveform will be altered on the first and last period of the sine wave only. If illuminated by a Walsh wave, the effect of the other target is seen during the entire period of the pulse. This seems to indicate superior resolution capability for the Walsh wave.

8. Information Content.

A sine wave has vanishingly little information content. Give me four points without violating the sampling theorem and I'll give you the whole sine wave. A Walsh function, on the other hand has the property that the more samples you get, the more information you have about it. You may have to think about that for a while. I know I did.

9. Speech synthesis and analysis.

Two of my graduate students have used Haar Functions in connection with speech. The first of these was a speech synthesis, redundancy-removal effort conducted by Meltzer.[7] The motivation was due to the fact that since hard-clipped speech is still recognizable, the information may be contained in the zero-crossings. The Haar Functions are extremely well suited to a zero-crossing context. Briefly, Meltzer's results indicate that with sampling at 5kHz and a subsequent Haar Transform, at least one half of the coefficients may be eliminated without loss of intelligibility on playback. It is theorized that up to one half of the remainder may also be removed, but the question is which one-half. It seems to vary depending upon the sounds present. Current investigation into analysis using Haar Functions is just starting. It is too soon to give results.

So what's next? As far as I'm concerned, the application of Walsh functions as well as other square-wave functions such as Haar functions is a wide-open field. As evidenced by the broad subject list from the several Walsh

Function Workshops [6] the interest is becoming more general, and the applications more diverse. I cannot say "What's next". I can only see a lot of interesting times ahead for me, my graduate students, and, perhaps some of you.

Bibliography

1. Harmuth, H.F., Transmission of Information by Orthogonal Functions, Springer-Verlag, Berlin/New York, 1970.
2. Pratt, W.K., et al, "Hadamard Transform Image Coding," Proceedings of the IEEE, vol. 57, no. 1, January, 1969.
3. Harmuth, H.F., "Sequence Filters Based on Walsh Functions," IEEE Transaction on Electromagnetic Compatibility, EMC, vol. 10, 1968.
4. Decker, J.A., and Harwit, M., "Experimental Operation of a Hadamard Spectrometer," Applied Optics, vol. 8, no. 12, Dec. 1969.
5. Ballard, H.A., "Telemetry Multiplexing with Orthogonal Pulse Waveforms," Proceedings of the National Telemetry Conference, 1963.
6. Bass, C.A., editor, "Applications of Walsh Functions," Proceedings of the Symposium and Workshop, 1970.
7. Meltzer, D., "Speech Recognition based on the Haar System of Orthogonal Functions," M.Sc. thesis, Ohio State Univ., 1969.

WALSH FUNCTIONS AND HADAMARD TRANSFORM

N. Ahmed

Depts. of Elec. Engg. & Comp. Sci.

Kansas State University

Manhattan, Kansas 66502

K. R. Rao

Dept. of Elec. Engg.

Univ. of Texas at Arlington

Arlington, Texas 76010

Abstract

This is a tutorial paper which deals with the fundamental aspects of the transform properties of Walsh functions. By appropriately sampling Walsh functions, it is shown that one can generate a class of matrices which have transform properties. These matrices are called Hadamard matrices. The name of the corresponding transform is the Walsh-Hadamard, or Hadamard transform (WHT or HT).

The Walsh-Hadamard analysis resembles the Fourier harmonic analysis in both geometrical and analytical characteristics. While the Fourier bases are sinusoids with harmonic frequencies, the Walsh-Hadamard bases are Walsh functions. Since Walsh functions are square waves, the corresponding Hadamard matrices consist of elements which are ± 1 and hence enable relatively easy information processing in several applications.

Introduction

Walsh functions¹ and their transforms² have found recently a number of applications in diverse areas.³⁻¹⁴ The symposium on applications of Walsh functions¹⁵ held annually is a testimony to this. The object of this paper is to present a tutorial exposition on the transform properties of Walsh functions.

Walsh Functions

Walsh functions, which form a complete orthonormal set of rectangular waves, can be defined several ways.^{1,6} The definition⁵ adopted here is based on Rademacher functions,⁷ which form an incomplete orthonormal set, and Gray code. Rademacher functions, $R_n(t)$ are periodic rectangular pulses, with 2^{n-1} cycles on the unit interval and alternate between $+1$ and -1 , with the exception of $R_0(t)$ which is a unit step function (Fig. 1). Walsh functions, $Wal(m, t)$ are developed as products of the Rademacher functions based on the Gray code, i. e., convert m to binary, $m = (b_n b_{n-1} \dots b_2 b_1)$ binary, then the Gray code of m is $(g_n g_{n-1} \dots g_2 g_1)$ binary, where

$$\begin{aligned} g_n &= b_n \\ g_{n-1} &= b_n \oplus b_{n-1} \\ g_2 &= b_3 \oplus b_2 \\ g_1 &= b_2 \oplus b_1 \end{aligned}$$

where the symbol \oplus denotes addition modulo 2.

Then

$$Wal(m, t) = (R_n(t))^{g_n} (R_{n-1}(t))^{g_{n-1}} \dots (R_2(t))^{g_2} (R_1(t))^{g_1}$$

For example, when $m=6=(110)$ binary, its Gray code is (101) binary, and hence

$$Wal(6, t) = (R_3(t)) (R_1(t)).$$

The first set of eight Walsh functions with corresponding sine and cosine harmonics superimposed on them is shown in Fig. 2. Unlike sinusoidal functions, the interval between successive zero crossings may not be the same for a given Walsh function. The term analogous to frequency is sequency,¹¹ which is based on the number of sign changes, g . Sequency is one half the number of zero crossings per unit interval (zero crossing at the left end only and not at the right end is counted.). In terms of sign changes, sequency, s , is defined as

$$\begin{aligned} s &= (g+1)/2, \quad g \text{ odd,} \\ s &= g/2, \quad g \text{ even.} \end{aligned}$$

Walsh functions can be further classified as "sal" (sine-WALSH) or "cal" (cosine-WALSH) based on the odd or even symmetry of $Wal(m, t)$ at $t=1/2$, i. e.,

$$\begin{aligned} Wal(2n-1, t) &= \text{sal}(m, t) \\ Wal(2n, t) &= \text{cal}(m, t) \end{aligned} \quad (\text{sequency}=m)$$

Discrete Walsh Transform

The evolution of FFT (Fast Fourier Transform) and the digital computers have brought digital signal and image processing and digital communications into limelight. The discrete version of the Walsh transform is based on the sampled values of the Walsh functions and the sampled data of the time series. The discrete Walsh transform¹⁴ of an N -periodic sampled data $x(k)$, $k=0, 1, \dots, N-1$ and $n=\log_2 N$ is given by

$$\{X(n)\} = \frac{1}{N} \{W(n)\} \{x(n)\}, \quad (1)$$

where $\{x(n)\}$ is the N -dimensional data (column) vector, $\{X(n)\}$ is the N -dimensional Walsh transform (column) vector, and $\{W(n)\}$ is the $(2^n \times 2^n)$ Walsh matrix. As $\{W(n)\}$ is orthogonal and symmetric, the inverse Walsh matrix is

$$\{x(n)\} = \{W(n)\} \{X(n)\}. \quad (2)$$

The transform vector $\{X(n)\}$ represents the frequency decomposition of $\{x(n)\}$, whereas the DFT (discrete Fourier transform) decomposes the data vector into its frequency components.

Hadamard Transform

The rows of the Walsh matrix can be rearranged to obtain Hadamard matrix^{4,16} which has a simple recursive structure. This rearrangement is based on the following simple rule:

Row of Walsh matrix, $m=(b_n b_{n-1} \dots b_2 b_1)$ binary, Gray code of m is $(g_n g_{n-1} \dots g_2 g_1)$ binary

Row of Hadamard matrix, $l=(g_1 g_2 \dots g_n g_{n-1})$ binary, which is the bit reversal of the Gray code of m . For example let $m=2=(010)$ binary, Gray code of 2 is (011) binary whose bit reversal is (110) binary = 6. For $N=8$, the Walsh matrix based on the sampled data of the Walsh functions, and the corresponding Hadamard matrix are,

$$\{x(n)\} = [H(n)] \{B_x(n)\}, \quad (5)$$

where $\{B_x(n)\}$ is the N -dimensional Hadamard transform (column) vector.

Fast Hadamard Transform

By factoring $[H(n)]$ into n sparse matrices, an efficient algorithm for fast computation of the transform can be developed.^{2,13} For example, $[H(3)]$ can be factored as

$$[H(3)] = [H_1(3)] [H_2(3)] [H_3(3)]$$

where

$$[H_1(3)] = \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ & & 1 & 1 \\ & & 1 & -1 \\ & & & & 1 & 1 \\ & & & & 1 & -1 \\ & & & & & & 1 & 1 \\ & & & & & & 1 & -1 \end{bmatrix},$$

Sampled data of Walsh function

Wal(0, t)
Wal(1, t) = sal(1, t)
Wal(2, t) = cal(1, t)
Wal(3, t) = sal(2, t)
Wal(4, t) = cal(2, t)
Wal(5, t) = sal(3, t)
Wal(6, t) = cal(3, t)
Wal(7, t) = sal(4, t)
and

$$[W(3)] = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$$

Row of $[H(3)]$

Number of sign changes

Sequency

0	0	0
4	1	1
6	2	1
2	3	2
3	4	2
7	5	3
5	6	3
1	7	4

Wal(0, t)
Wal(7, t)
Wal(3, t)
Wal(4, t)
Wal(1, t)
Wal(6, t)
Wal(2, t)
Wal(5, t)

$[H(3)] =$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

Row of $[W(3)]$

Number of sign changes

Sequency

0	0	0
7	7	4
3	3	2
4	4	2
1	1	1
6	6	3
2	2	1
5	5	3

Hadamard matrix of any order can be generated recursively from

$[H(0)] = 1$, and

$$[H(K+1)] = \begin{bmatrix} [H(K)] & [H(K)] \\ [H(K)] & -[H(K)] \end{bmatrix}, \quad K = 0, 1, 2, \dots \quad (3)$$

$$= [H(1)] \otimes [H(K)]$$

where $[H(1)] = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ and the symbol \otimes denotes Kronecker product. Hadamard matrix is both symmetric and orthogonal, i. e., $[H(n)] [H(n)]^T = NI_N$, where I_N is the $(N \times N)$ identity matrix. The Hadamard transform (HT) of $\{x(n)\}$ and its inverse are respectively defined as,^{2,4}

$$\{B_x(n)\} = \frac{1}{N} [H(n)] \{x(n)\}, \quad (4)$$

and

$$[H_2(3)] = \begin{bmatrix} I_2 & I_2 \\ I_2 & -I_2 \end{bmatrix}$$

and

$$[H_3(3)] = \begin{bmatrix} I_4 & I_4 \\ I_4 & -I_4 \end{bmatrix}. \quad (6)$$

The signal flow graph based on (6) is shown in Fig. 3. The fast algorithm requires about $N \log_2 N$ arithmetic operations (Hadamard transform involves additions and subtractions only as all the elements in the transform matrix are +1 or -1) compared to N^2 required by the direct method. Factoring of transform matrix of any order and developing the corresponding flow graph is straightforward.

Power Spectrum

The power spectrum invariant to the cyclic shift of the sampled data can be developed using the shift matrix. If $\{x^{(l)}(n)\}$ is $\{x(n)\}$ shifted cyclically to the left by l places and if $\{B_x^{(l)}(n)\}$ is the HT of the shifted sequence $\{x^{(l)}(n)\}$, then

$$\begin{aligned} \{B_x^{(l)}(n)\} &= \frac{1}{N} [H(n)] \{x^{(l)}(n)\} \\ &= \frac{1}{N} [H(n)] [M^{(l)}(n)] \{x(n)\} \quad (7) \\ &= \frac{1}{N} [H(n)] [M^{(l)}(n)] [H(n)] \{B_x(n)\} \end{aligned}$$

where $[M^{(l)}(n)]$ is I_N whose columns are shifted to the right cyclically by l places. The transforms of the shifted and the original sequences are related through the shift matrix, $[S^{(l)}(n)]$, i. e.,

$$\{B_x^{(l)}(n)\} = [S^{(l)}(n)] \{B_x(n)\}, \quad (8)$$

where $[S^{(l)}(n)] = \frac{1}{N} [H(n)] [M^{(l)}(n)] [H(n)]$ represents the similarity transformation.

The shift matrix has block diagonal orthonormal structure. For $N = 8$, and $l = 1$, the shift matrix becomes

$$[S^{(1)}(3)] = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & 1 \end{bmatrix} \quad (9)$$

where each of the block diagonal matrices is orthonormal. This is the key to the invariance of the power spectrum to the periodic shift of the time series. Also the power spectrum for $N = 8$ is as follows:

$P_0 = B_x^2(0)$	Sequence	0	
$P_1 = B_x^2(1)$		4	
$P_2 = B_x^2(2) + B_x^2(3)$		2	(10)
$P_3 = \sum_{K=4}^7 B_x^2(K)$		1, 3	

In general for any $N = 2^n$, the power spectrum and its sequence composition is as follows:

$P_0 = B_x^2(0)$	Sequence	0	
$P_1 = B_x^2(1)$		$\frac{N}{2}$	
$P_2 = \sum_{K=2^{n-1}}^{2^2-1} B_x^2(K)$		$\frac{N}{4}$	(11)
$P_3 = \sum_{K=2^{3-1}}^3 B_x^2(K)$		$\frac{N}{8}, \frac{3N}{8}$	

$P_n = \sum_{K=2^{n-1}}^{2^n-1} B_x^2(K)$	Sequence	$1, 3, 5, 7, \dots, (\frac{N}{2} - 1)$
---	----------	--

An inspection of (11) shows that the Hadamard power spectral points do not represent individual sequences but groups of sequences. But this grouping is not arbitrary, but is based on the half-wave symmetry structure, i. e., each group contains a fundamental and all the odd harmonics relative to that fundamental.

Modified Hadamard Transform

The HT power spectrum (11) can be computed directly using the modified Hadamard transform (MHT). The MHT of $\{x(n)\}$ and its inverse are respectively defined as,

$$\{F(n)\} = \frac{1}{N} [D(n)] \{x(n)\}, \quad (12)$$

and

$$\{x(n)\} = [D(n)] \{F(n)\}, \quad (13)$$

where $\{F(n)\}$ is the N -dimensional MHT (column) vector. The transform matrix $[D(n)]$ is orthogonal and can be generated recursively as follows:

$$[D(K+1)] = \begin{bmatrix} [D(K)] & [D(K)] \\ 2^{K/2} I_{2K} & -2^{K/2} I_{2K} \end{bmatrix}, \quad K = 0, 1, 2, \dots \quad (14)$$

with $[D(0)] = 1$. For $N = 8$, the transform matrix is,

$$[D(3)] = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} & 0 & \sqrt{2} & 0 & -\sqrt{2} & 0 \\ 0 & \sqrt{2} & 0 & -\sqrt{2} & 0 & \sqrt{2} & 0 & -\sqrt{2} \\ \hline & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \end{bmatrix} \quad (15)$$

and the flow graph for fast computation of $\{F(3)\}$ is shown in Fig. 4. As in the case of HT, the flow graph is based on factoring $[D(3)]$ into sparse matrices i. e.,

$$[D(3)] = [D_1(3)] [D_2(3)] [D_3(3)]$$

where

$$[D_1(3)] = \begin{bmatrix} 1 & 1 & & & & & & \\ & 1 & -1 & & & & & \\ & & & \sqrt{2} I_2 & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \end{bmatrix}, \quad (16)$$

$$[D_2(3)] = \begin{bmatrix} I_2 & I_2 \\ I_2 & -I_2 \\ \hline & & 2 I_4 \end{bmatrix},$$

and

$$[D_3(3)] = \begin{bmatrix} I_4 & I_4 \\ I_4 & -I_4 \end{bmatrix}.$$

The MHT requires much less number of arithmetic operations compared to HT, because of the sparseness in $[D(n)]$ and consequent increased sparseness in its factors. Comparison of Figures 3 and 4 yields the following relationships between the BT and MBT coefficients:

$$B_x(0) = F(0), B_x(1) = F(1),$$

$$\begin{Bmatrix} B_x(2) \\ B_x(3) \end{Bmatrix} = \frac{1}{\sqrt{2}} [H(1)] \begin{Bmatrix} F(2) \\ F(3) \end{Bmatrix},$$

and

$$\begin{Bmatrix} B_x(4) \\ B_x(5) \\ B_x(6) \\ B_x(7) \end{Bmatrix} = \frac{1}{2} [H(2)] \begin{Bmatrix} F(4) \\ F(5) \\ F(6) \\ F(7) \end{Bmatrix}.$$

From (10) and (17) it is clear that the BT power spectrum can also be expressed as,

$$\begin{aligned} P_0 &= B_x^2(0) = F^2(0) \\ P_1 &= B_x^2(1) = F^2(1) \\ P_2 &= \sum_{k=2}^3 B_x^2(k) = \sum_{k=2}^3 F^2(k) \\ P_3 &= \sum_{k=4}^7 B_x^2(k) = \sum_{k=4}^7 F^2(k) \end{aligned} \quad (18)$$

or in general,

$$P_0 = B_x^2(0) = F^2(0) \quad (19)$$

$$F_m = \sum_{k=2^{m-1}}^{2^m-1} B_x^2(k) = \sum_{k=2^{m-1}}^{2^m-1} F^2(k), m = 1, 2, \dots, n.$$

The BT power spectrum can thus be evaluated directly without computing $\{B_x(n)\}$.

Multidimensional Hadamard Transform

As with other discrete orthogonal transforms, HT can be extended to multiple dimensions. For example, the two dimensional HT, which has been used in image processing, can be defined as

$$B_x(u_1, u_2) = \frac{1}{N_1 N_2} \sum_{v_1=0}^{N_1-1} \sum_{v_2=0}^{N_2-1} x(v_1, v_2) (-1)^{\langle v, u \rangle} \quad (20)$$

where

$B_x(u_1, u_2)$ is the transform coefficient,

$x(v_1, v_2)$ is the sampled data,

$$v_1, u_1 = 0, 1, 2, \dots, N_1-1, \quad n_1 = \log_2 N_1,$$

$$v_2, u_2 = 0, 1, 2, \dots, N_2-1, \quad n_2 = \log_2 N_2.$$

$$\langle v, u \rangle = \langle v_1, u_1 \rangle + \langle v_2, u_2 \rangle,$$

$$\langle v, u \rangle = \sum_{m=0}^{n_2-1} v_2(m) u_2(m), \quad t = 1, 2.$$

The terms $v_2(m)$ and $u_2(m)$ are the binary representations of v_2 and u_2 respectively, i. e.,

$$(v_1)_{\text{decimal}} = (v_1(n_1-1) v_1(n_1-2) \dots v_1(1) v_1(0))_{\text{binary}}.$$

The inverse transform of (20) is

$$x(v_1, v_2) = \sum_{u_1=0}^{N_1-1} \sum_{u_2=0}^{N_2-1} B_x(u_1, u_2) (-1)^{\langle u, v \rangle} \quad (21)$$

Conclusions

A tutorial exposition on Walsh functions and their transforms is presented. The transform analysis based on Walsh functions amounts to sequency decomposition of a signal or image unlike frequency decomposition in the case of Fourier transform. As only additions and subtractions are involved in evaluating the Hadamard transform, savings in execution time and memory requirements of the digital computer can be gained compared to DFT. However, the HT power spectral points represent groups of sequencies unlike the individual frequency representation in the case of DFT.

References

1. J. L. Walsh, "A Closed Set of Orthogonal Functions," Amer. J. of Mathematics, vol. 45, pp. 5-24, 1923.
2. N. Ahmed, R. Rao and A. L. Abdusattar, "BIFORE or Hadamard Transform," IEEE Trans. Audio and Electroacoustics, vol. AU-19, Sept. 1971.
3. J. E. Whelchel et al. "The Fast Fourier-Hadamard Transform and Its Use in Signal Representation and Classification," Aerospace Electronic Conference, EASCON Record, pp. 561-573, Sept. 9-11, 1968.
4. W. K. Pratt, et al. "Hadamard Transform Image Coding," Proc. IEEE, vol. 57, pp. 58-68, Jan. 1969.
5. N. Ahmed and K. R. Rao, "Spectral Analysis of Linear Digital Systems Using BIFORE," Electronics Letters, vol. 6, pp. 43-44, Jan. 22, 1970.
6. H. F. Harmuth, "Applications of Walsh Functions in Communications," IEEE Spectrum, vol. 6, pp. 82-91, Nov. 1969.
7. M. S. Corrington, "Advanced Analytical and Signal Processing Techniques," ASTIA Document No. AD-277942, April 1962.
8. R. B. Lackey, "So What's a Walsh Function," Proc. IEEE Fall Electronics Conference, Oct. 18-20, 1971, Chicago, Ill., pp. 368-371.

9. D. M. Walsh, "Design Considerations for Digital Walsh Filters," *ibid.*, pp. 372-375.
10. M. F. Harmuth, "Sequence Filters Based on Walsh Functions," *IEEE Trans. Electromagnetic Compatibility*, vol. EMC-19, pp. 293-295, June 1966.
11. H. F. Harmuth, "Transmission of Information by Orthogonal Functions," Springer-Verlag, II Printing, 1979.
12. T. S. Huang, et al, "Image Processing," *Proc. IEEE*, vol. 59, pp. 1586-1609, Nov. 1971.
13. H. C. Andrews and K. L. Caspari, "A Generalized Technique for Spectral Analysis," *IEEE Trans. Computers*, vol. C-19, pp. 16-25, Jan. 1970.
14. Proceedings of the Symposium on Applications of Walsh Functions, March 31-April 4, 1971 and April 13-15, 1971, Washington, D. C., Documents #s (1970) AD707431 and (1971) AD727000, National Technical Information Service, Operations Division, Springfield, Va. 22151.
15. R. B. Lackey and D. Meltzer, "A Simplified Definition of Walsh Functions," *IEEE Trans. Computers*, vol. C-20, pp. 211-213, Feb. 1971.
16. K. W. Henderson, "Some Notes on the Walsh Functions," *IEEE Trans. Electronic Computers*, vol. EC-13, pp. 50-52, Feb. 1964.
17. N. Ahmed and R. B. Schultz, "Position Spectrum Considerations," *IEEE Trans. Audio and Electroacoustics*, vol. AU-19, pp. 326-327, Dec. 1971.

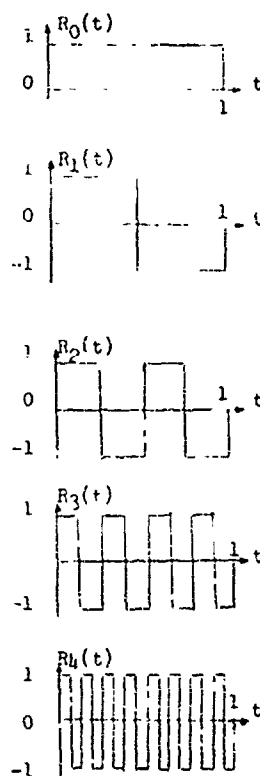


Fig. 1. Padenacher functions

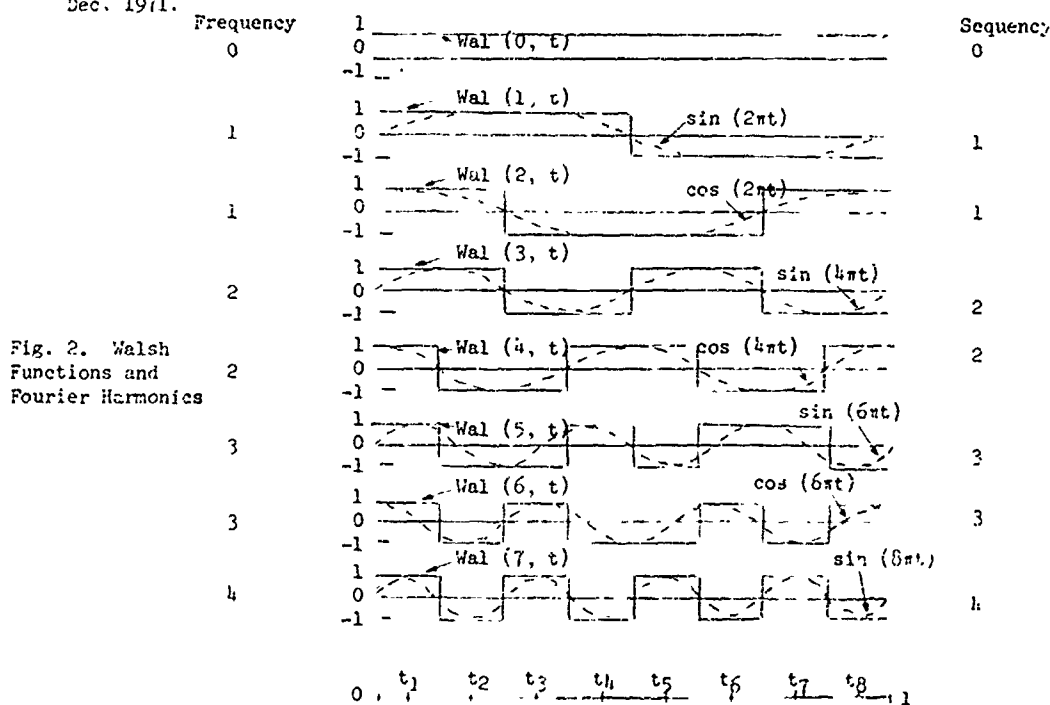


Fig. 2. Walsh Functions and Fourier Harmonics

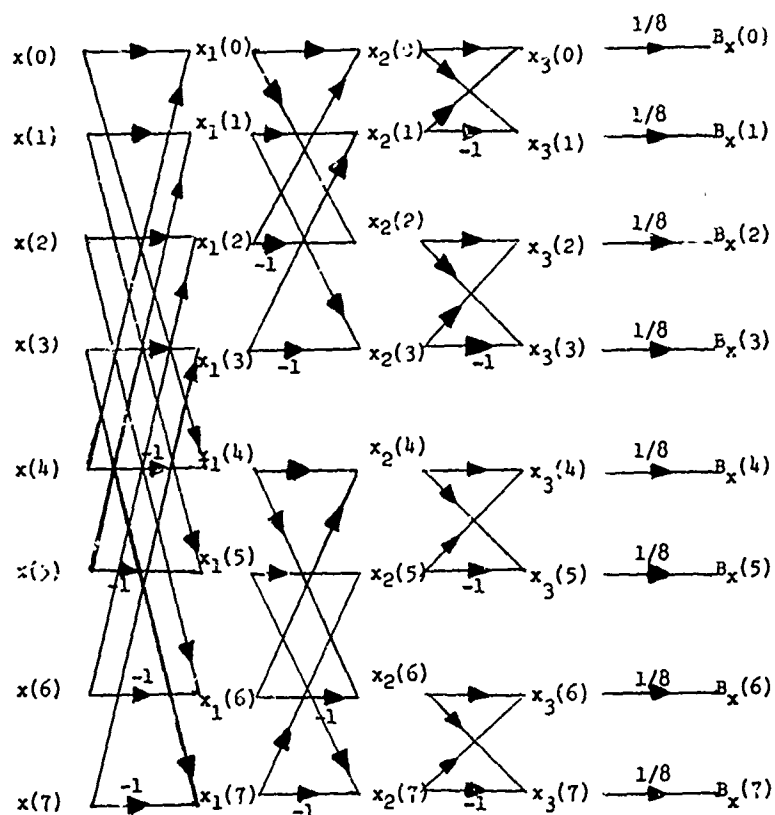


Fig. 3. Signal flow graph illustrating the computation of HT for $N = 8$.

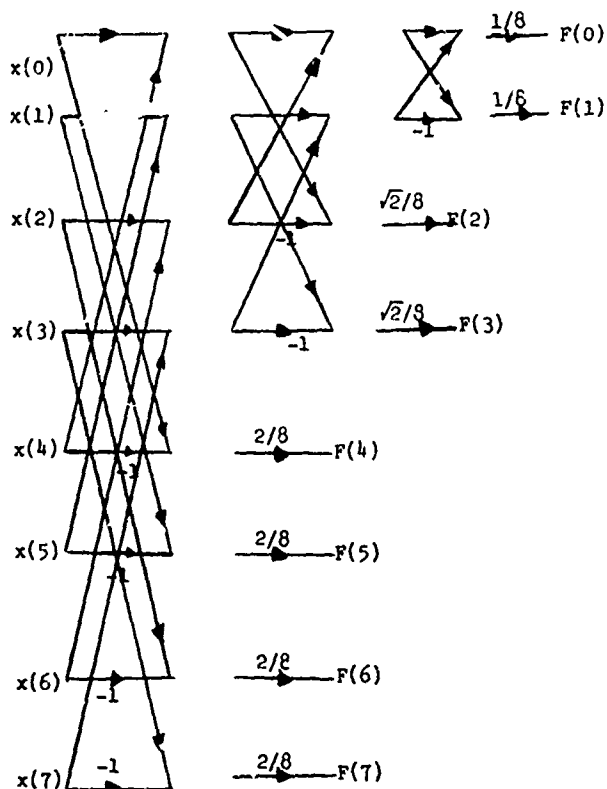


Fig. 4. Signal flow graph for the modified HT for $N = 8$.

Walsh Functions in Image Processing and Two-Dimensional Filtering*

William K. Pratt
Department of Electrical Engineering
University of Southern California
Los Angeles, California

INTRODUCTION

The role of Walsh Functions in image Processing and Two Dimensional Filtering has been, appropriately enough, binary. Walsh Functions have found useful application for image coding, image enhancement, pattern recognition, and general two dimensional filtering [1,4]. In addition, the introduction of Walsh functions to these applications has fostered new concepts of generalized spectral analysis and signal representation. Furthermore, research on Walsh functions has led to a better understanding of images and their structure than was previously available from purely sine wave analysis.

Digital Images

Before embarking on the subject of Walsh function image processing, it will be fruitful to present some background on digital image processing.

Image Representation

A digital monochrome image, denoted as $f(x, y)$, is defined here as an array (assumed square of dimension N by N for simplicity) of samples of a continuous two dimensional intensity pattern of light. Each picture element (pixel) of $f(x, y)$ is assumed to be limited in amplitude to A units and linearly quantized to L levels where L is usually a binary number, i. e. $L=2^a$ with a an integer. Since $f(x, y)$ represents samples of light intensity, the image array must be positive. It can be further argued that every intensity pattern contains at least a few quanta of light. Hence, it will be assumed that

$$0 < f(x, y) \leq A$$

In the past few years there has been an expanding activity in the processing of digital color imagery. The discussion here will be limited to a consideration of digital color images represented by the three primary color systems. In this system,

which is the basis for color television systems and photographic processes, the color image is described at each coordinate point by three arrays $f_R(x, y)$, $f_G(x, y)$, $f_B(x, y)$ that specify the red, green, and blue tristimulus values of an image point. The tristimulus values designate the amounts of red, green, and blue primary lights that are required to provide a colorimetric match of a color by a display system.

Image Processing Systems

Figure 1a contains a block diagram of a general image processing system based upon the processing of the image intensity. Many researchers have suggested that the logarithm of the intensity, which is proportional to photographic density, should be processed or coded in a system, such as shown in figure 1b, rather than the image intensity [4]. The rationale behind this suggestion, indicated in figure 1c, is that the human eye responds logarithmically to intensity, and actually is a linear system to logarithmic changes in brightness. The performance of the image "density" processor has been found to be superior to the image "intensity" processor in many applications. However, it should be remembered that both systems are predicated upon an ideal sensor intensity response and an ideal image intensity display. If the sensor or display responds nonlinearly, extreme image degradation may occur. Most physical image sensors and displays are inherently nonlinear in intensity, and it is necessary to compensate for the nonlinearity either electrically or photographically if precision image processing is to be performed.

Sampling and Quantization

In the analysis of a digital image processing system, consideration should be given to the effects of sampling and quantization. The two dimensional sampling may not necessarily be performed at a sufficient rate to satisfy the two dimensional sampling theorem. The implications of processing an under or over sampled image

*This research was supported by the Advanced Research Projects Agency of the Defense and was monitored by the Air Force Eastern Test Range under Contract No. F08606-72-C-0008.

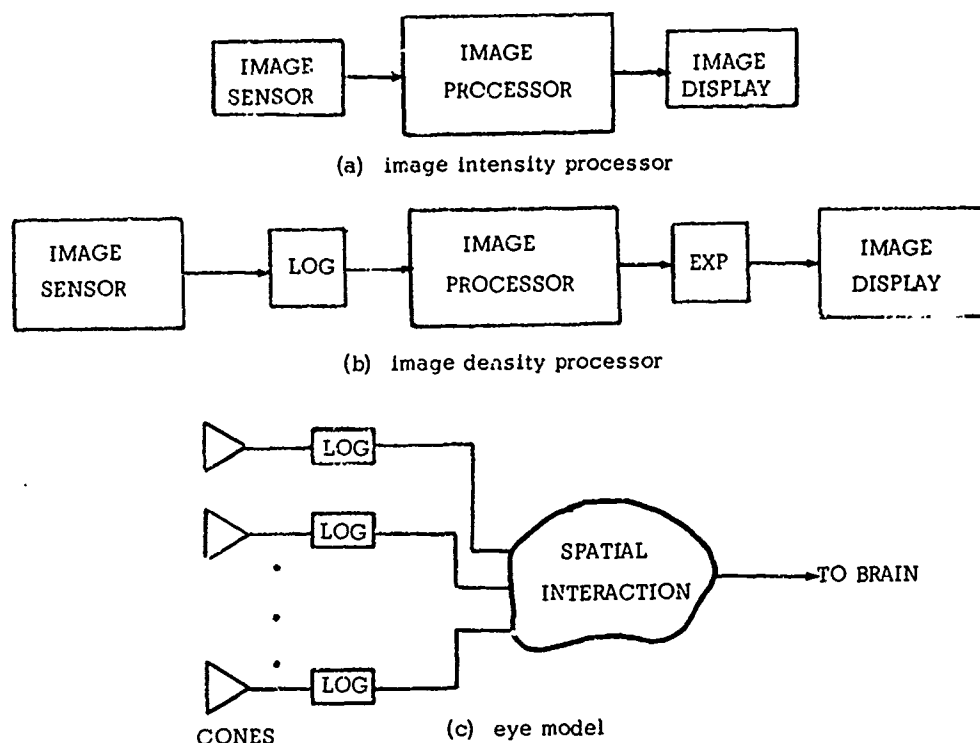


Figure 1. Image processing systems

are indeed serious, and can cloud the expected processing results. If the image is under sampled, spurious low spatial frequency components may be unintentionally introduced. On the other hand, an over sampled image contains redundant samples which unnecessarily increase the computational load. Also, it should be realized that quantization introduces a non-recoverable error in the specification of the amplitude of each image sample. The number of quantization levels required to maintain the quantization error below the subjective threshold of noticeability is strongly dependent upon the characteristics of the image sensor and display. It is obviously not worthwhile to quantize too finely if the image sensor is noisy or if the image display is incapable of rendering only a few grey shades. While the eye is only capable of the absolute discrimination of 10 to 15 shades of grey, it can detect the relative brightness between two grey shades with much greater sensitivity. The effect of too few quantization levels is usually first noticed as grey scale contouring over regions of gradual shading. Generally speaking, 64 quantization levels are sufficient for television broadcast quality displays. Up to 256 quantization

levels may be required for a high quality mechanical or flying spot scanner display. These limits assume a linear quantization of the image intensity.

Performance Measures

In the development of image processing systems it is highly useful, and often necessary, to have some quantitative performance measures. The search for a quantitative measure of image quality has been long, and unfortunately, not particularly fruitful. For monochrome imagery because of its simplicity, most researchers have utilized the mean square error between an original image $f(x, y)$ and a processed image $\tilde{f}(x, y)$ as defined by

$$e = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} [f(x, y) - \tilde{f}(x, y)]^2 \quad (2)$$

where the overbar indicates a statistical overage. It has been suggested that the basic mean square error expression can be improved by taking into consideration the

impulse response $h(x, y)$ of the human eye [5]. The resultant error is given by

$$e = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} [f(x, y) * h(x, y) - \tilde{f}(x, y) * h(x, y)]^2 \quad (3)$$

where $*$ indicates a spatial convolution. It is possible to extend both of these image error measures to each of the tristimulus arrays of a color image. Neither the ordinary nor spatial frequency weighted mean square error expression has proven to be completely adequate measure of monochrome or color image quality; much more research effort is needed in this area.

For image coding it is necessary to measure the redundancy reduction or bandwidth compression obtained by a coding process as well as specify any possible degradation in image quality. The specification of image coding performance should be a relatively simple matter, but seems to be a subject of much confusion in the literature. There are two "compression factors" that are often stated as image coding performance measures:

$$\text{sample reduction factor} = \frac{\text{number of original image samples}}{\text{number of coded image samples}}$$

$$\text{bit reduction factor} = \frac{\text{number of original image code bits}}{\text{number of coded image code bits}}$$

The sample and bit reduction factors are identical if the same number of bits are assigned to each sample. Both measures, however, can be misleading since they do not indicate the "information" content of the original image. A high compression factor can often be obtained if the original image is over-sampled and over quantized. The information content of an image can, in theory, be measured by the entropy of the image source. But, in general, the computation is difficult. An estimate of the total source entropy can be obtained by performing a grey scale histogram of the image and computing the first order entropy as given by

$$H\{f(x, y)\} = - \sum_{k=1}^L P_1(k) \log_2 [P_1(k)] \quad (4)$$

where

$$P_1(k) = P_r \{f(x, y) = k\}$$

is the image amplitude probability distribution.

Image Transformation

The two dimensional Hadamard transform $F(u, v)$ of an image function $f(x, y)$ can be written in series form as

$$F(u, v) = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) (-1)^{q(x, y, u, v)} \quad (5)$$

where $q(x, y, u, v)$ is defined in [1]. It is often more convenient to express the transformation in matrix notation as

$$[F] = [H_N] [f] [H_N] \quad (6)$$

where $[H_N]$ is an N th order real, symmetric, and normalized Hadamard matrix whose rows are Walsh functions. The Hadamard matrix $[H_N]$ may be obtained by the construction

$$[H_2] = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$[H_N] = \frac{1}{\sqrt{2}} \begin{bmatrix} H_{N/2} & H_{N/2} \\ H_{N/2} & -H_{N/2} \end{bmatrix} \begin{bmatrix} P_N \\ \end{bmatrix}$$

where the matrix $[P_N]$ is a permutation matrix that orders the sequences of the Walsh functions. The computation indicated by equation (6) is simply a one dimensional Hadamard transform of each row of the image array and a one dimensional transform of each column of the resultant semi-transformed array. If the image array is too large to be stored in the computer then it can be placed in a secondary sequential storage unit (magnetic tape or disc) and brought to the computer a line at a time. The computation then proceeds as follows:

$$[F_1] = [f] [H_N] \quad \text{- row transform}$$

$$[F_2] = [F_1]^T \quad \text{- matrix transpose}$$

$$[F_3] = [F_2] [H_N] \quad \text{- row transform}$$

$$[F] = [F_3]^T \quad \text{- matrix transpose}$$

The last matrix transposition usually can be ignored since it is immaterial whether one transforms the original image, or its transpose. The first matrix transposition can be performed efficiently by the algorithm described in reference [6].

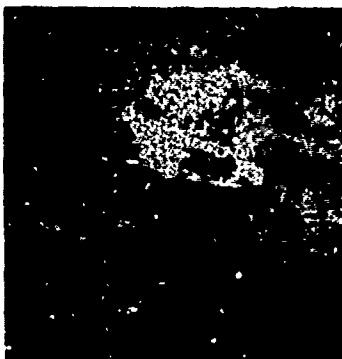
Figure 2 contains photographs of two original images and magnitude displays of the



(a) Original girl



(b) Hadamard display of girl



(c) Original tank



(d) Hadamard display of tank

Figure 2. Original images and transforms

Hadamard transforms. The images contain 256 by 256 pixels and 64 grey levels.

Image Coding

The Hadamard transform has received much attention for its use in image coding applications to reduce channel bandwidth. By itself, the transformation does not reduce channel capacity requirements because the transformation is information preserving. That is, the entropy of the image and the entropy of the transform of the image are identical. A reduction in the channel coding requirements, however, can be obtained for natural images, at the expense of some average distortion, by efficiently coding the Hadamard transform domain coefficients for transmission. It is important to remember that the transform domain coding process will usually introduce some average distortion, and that there is no effective means of limiting the peak distortion. Thus, application of transform coding is often limited to image transmission systems for human viewing.

In the transform coding process a two dimensional Hadamard transform is taken over the full size image or some smaller segment called a block. Next each transform coefficient is quantized and coded. At the receiver, after decoding, an inverse transform is taken to reconstruct the image. Two basic methods of transform domain image coding have been applied. The first, called zonal coding, entails the establishment of zones in the transform domain. In each zone every coefficient is then quantized, usually according to some nonlinear amplitude scale, and assigned a code group. A more efficient variant of zonal coding is to assign an average code length to each zone and use variable length Huffman coding of the quantized samples. In either case, the bit assignment is based upon the assumed variance of the transform domain samples. A convenient algorithm, based upon results of rate distortion theory, is to select the number of bits according to the relation

$$N_B(u, v) = \ln [\sigma_F^2(u, v)] - \ln [D] \quad (7)$$

where $\sigma_F^2(u, v)$ is the variance of a transform domain coefficient and D is proportional to the mean square error of the coding process. Figure 3 illustrates a typical assignment of code bits for image coding in 16 by 16 pixel blocks. The performance of the transform zonal coding system can be specified in terms of the mean square error between

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

MAXIMUM VARIANCE
ZONAL FILTER
4:1 SAMPLE REDUCTION

$X_c = .95$
 $Y_c = .95$

MEAN SQUARE ERROR

FOURIER TRANSFORM

HAAR TRANSFORM

HADAMARD TRANSFORM

K-L TRANSFORM
(correl. coeff. = .95)

SLANT TRANSFORM

BLOCK VALUE

2x2 4x4 8x8 16x16 32x32 64x64 128x128 256x256

Figure 4 contains a plot of mean square error as a function of block size for several image transforms. This plot was obtained for an image statistically described by a Markov process. In the coding process the transform coefficients are ranked according to their variance. The 25% with the largest variance are coded with 6 bits each and the remainder are discarded. Thus, the coding requires an average of 1.5 bits/pixel. From the diagram it is seen that the Hadamard transform compares favorably with the other transforms, which are more complicated to implement, and provides a coding of less than 1% mean square error for reasonably small block sizes.

The second type of Walsh domain coding, called threshold coding, is based upon the establishment of a magnitude threshold. If the Walsh coefficient is greater than the threshold it is quantized and coded, otherwise it is discarded. It is necessary, of course, to code the location of the significant coefficients as well as their amplitude. The performance of the threshold coder is somewhat better than that of the zonal coder, but is more difficult to implement. Figure 5 contains reconstructions



(a) Zonal coding 1.5 bits/pixel



(b) Threshold coding 1.5 bits/pixel

18

of images coded with 1.5 bits/pixel for the zonal and threshold coding processes.

The concept of two dimensional transforms coding has been extended to color imagery. In this process the red, green, and blue tristimulus signals are linearly converted to the Y I Q color coordinate system where the Y signal represents the image luminance and the I and Q signals jointly provide the hue and saturation information of a color. Figure 6 contains photographs of the I and Q color planes of a typical color image. The Y plane is shown in figure 1a. It is seen that the IQ planes do not contain many high spatial frequencies, and hence can be subjected to a high degree of zonal coding without significant impairment. In a series of experiments it was found that the Y signal could be transform coded with an average of 3.0 bits/pixel and that the I and Q signals only required an average of 0.375 bits/pixel each for a total of 3.5 bits/pixel.



(a) I



(b) Q

Figure 6. I and Q tristimulus planes of a color image.

Two Dimensional Filtering

The initial concept of two dimensional Walsh function filtering to be developed here is very broad and generalized. Basically, the Walsh functions are used to perform a spectral decomposition of the image into a domain that facilitates the linear or nonlinear processing operations. The two dimensional filtering steps are indicated below:

$$f(x, y) \xrightarrow{\text{transform}} F(u, v) \xrightarrow[\text{process}]{\text{filter}} F(u, v) \xrightarrow{\text{transform}} f(x, y)$$

Linearity or nonlinearity of a filtering process is defined here by the operations performed on the Walsh domain components, not on the spatial domain elements prior to the forward transformation or after the inverse transformation.

Linear filtering. In its most general form linear filtering may be defined as a linear combination of all of the image transform domain components to produce a modified transform domain as defined by

$$F(u, v) = \sum_{u'=0}^{N-1} \sum_{v'=0}^{N-1} F(u', v') G(u, u', v, v') \quad (8)$$

where $G(u, u', v, v')$ is the filter weighting function. If the filter weighting function is separable

$$G(u, u', v, v') = G_u(u, u') G_v(v, v') \quad (9)$$

then the filtering operation can be performed as sequential operations on the rows and columns of the image transform. In this case it is convenient to switch to the matrix representation of the filtering operation.

$$[\tilde{F}] = [G_u][F][G_v] \quad (10)$$

If the filter matrices $[G_u]$ or $[G_v]$ are diagonal, then the filtering operation reduces to an individual weighting of the image frequency components. In general the optimum filter for a particular application is not diagonal since the frequency components are correlated.

Nonlinear filtering. In the definition of nonlinear Walsh domain filtering the only restriction that applies is that the filtering operation must perform a mapping of $F(u, v)$ into an array $\tilde{F}(u, v)$ of the same dimension. To date, the only nonlinear filtering functions

that have received much attention are the logarithmic

$$\widehat{F}(u, v) = \frac{F(u, v) \log \{ |F(u, v)| \}}{|F(u, v)|} \quad (11)$$

and power law

$$\widehat{F}(u, v) = \frac{F(u, v)}{|F(u, v)|^\alpha} |F(u, v)|^\alpha \quad (12)$$

functions where α is a constant. The logarithmic Walsh filtering operation is similar to the cepstrum operation associated with the Fourier transform.

Image Restoration and Enhancement

Image restoration has commonly been defined as the reconstruction of an image back to its original form as closely as possible. Typical restoration applications include correction for defocus and image motion blur, geometric distortion compensation, and noise filtering. Image enhancement entails operations that improve the appearance of an image to a human viewer. Several examples of the application of Walsh functions to image restoration and enhancement are given below.

Noise reduction a frequently encountered image processing problem is the reduction of the visual effects of image noise generated by the sensor, processing system, or display. Consider a "noisy" image, $f(x, y)$, composed of additive signal, $s(x, y)$, and noise, $n(x, y)$, components.

$$f(x, y) = s(x, y) + n(x, y) \quad (13)$$

The optimum filter to minimize the mean square error between the signal image $s(x, y)$ and the filtered image $\hat{f}(x, y)$ is obtained from an extension of classical Wiener filter theory [7]. For the special, but practical, case of the signal and noise uncorrelated with each other, and separate processing of the rows and columns of the image, the optimum filter matrices are found to be

$$[G_u]_{\text{opt}} = [H] [C_{sx} [C_{sx} + C_{nx}]^{-1} [H] \quad (14)$$

$$[G_v]_{\text{opt}} = [H] [C_{sy} [C_{sy} + C_{ny}]^{-1} [H] \quad (15)$$

where $[C_{sx}]$, $[C_{sy}]$, $[C_{nx}]$, $[C_{ny}]$ are the covariance matrices of the elements along the rows and columns of the signal and noise arrays, respectively. It is also

of great importance to note that the minimum mean square error obtained with Hadamard transform filtering is the same as could be obtained with the Fourier or Karhunen-Loeve transforms or any other unitary transformation. The potential advantage of using the Hadamard transform is the reduced amount of computation required as compared to many other transforms. Figure 7 contains displays of a Markov process covariance matrix for a vector length of sixteen and the corresponding optimum Hadamard Wiener filter matrix. It is possible to reduce the computational requirements even further by taking advantage of the character of the filter matrices. In many applications it will be found that the optimum filter matrices contain many near zero elements. If a filter element is very small it can simply be discarded from the computation. It has been found that up to 90% of the filtering multiplications can be avoided in this manner with only a slight increase in error for many applications.



Figure 7. Display of Hadamard Wiener filter for $N=16$ and Markov process signal with $R=0.9$

Figure 8 shows an example of Walsh domain Wiener filtering.

Inverse filtering. In many imaging systems an observable image can be considered as the output of a linear, shift invariant two dimensional system with an impulse response $h_d(x, y)$ that models the degrading mechanism of the imaging system. Thus, the degraded image is given by

$$f_d(x, y) = f(x, y) * h_d(x, y) \quad (16)$$

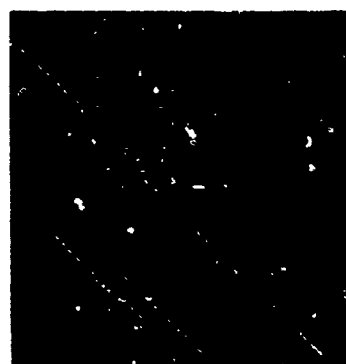
By the convolution theorem of Fourier transform theory



(a) Tank plus noise



(b) Hadamard Wiener filtering



(c) Selective computation Hadamard Wiener filtering

Figure 8. Example of Walsh domain Wiener filtering

$$\mathcal{F}_d(u, v) = \mathcal{F}(u, v) \mathcal{H}_d(u, v) \quad (17)$$

where the script letters indicate the two dimensional Fourier transforms of the degraded image, original image, and imaging system transfer function, respectively. If the Fourier spectrum is multiplied point by point with the transfer function $1/\mathcal{H}_d(u, v)$, the result is the original image Fourier spectrum, which can be inverse Fourier transformed to obtain the original image. This process of inverse Fourier image reconstruction is exact in theory, but often fails in practice because of the inversion

of the zeros of $\mathcal{H}_d(u, v)$ and amplification of high frequency image noise. An exact analog of Fourier inverse filtering does not exist for Walsh function because of the lack of an equivalent "Walsh convolution theorem" as represented by eqs (16) and (17). It is possible, however, to achieve some improvement in image quality by a form of Walsh domain inverse filtering in which the magnitude of the Fourier inverse filter is used as an inverse filter for the Walsh spectrum of an image. Reference 8 gives an example of Walsh domain inverse filtering.

Summary

When evaluating the usefulness of Walsh functions for image processing, one is led to the question: Do Walsh functions possess any properties inherently useful for image processing? The answer is equivocal. Walsh functions do not characterize an image particularly well; they do not "resemble" typical image lines. From this standpoint there is no reason to prefer Walsh functions over any other functions with a property of spectral decomposition. The major potential advantage of Walsh functions in image processing is their relatively simple computability. This is a major factor of consideration for the processing of pictures with millions of elements.

The future role of Walsh functions in image processing is uncertain to this observer. In image coding Hadamard transform coders may find practical use in specialized applications in which a small amount of image degradation is permitted, but the coder must be relatively simple. For the computer processing of images for restoration purposes, the Walsh functions hold promise. But, as mentioned earlier, perhaps the greatest benefit to the field of image processing to be derived from the introduction of Walsh functions is that they have stimulated research into new concepts of generalized spectral analysis, and have brought into deeper consideration the comprises of processing performance and computation requirements.

References

1. W. K. Pratt, H. C. Andrews, and J. Kane, "Hadamard Transform Image Coding," *Proc. IEEE*, Vol. 57, No. 1, January, 1969, pp. 58-68.

2. N.A. Alexandridis, "Walsh-Hadamard Transformations in Image Processing" Univ. of Calif. at Los Angeles, School of Engineering and applied Sciences, Report No. UCLA-ENG-7108, March, 1971
3. Proceedings of applications of Walsh Functions, Washington, D.C. for years 1969, 1970, 1971.
4. T.G. Stockham, "Intra-Frame Encoding for Monochrome Images by Means of a Psychophysical Model Based On Nonlinear Filtering of Multiplied Signals", Picture Bandwidth Compression, T.S. Huang, O. J. Tretiak, eds. Gordon and Breach, New York, 1972, pp. 415-442.
5. D. J. Sakrison and V.R. Algazi, "Comparison of Line-by-Line and Two-Dimensional Encoding of Random Images," IEEE Transactions on Information Theory, Vol. IT-17, No. 4, July, 1971, pp. 386-398.
6. J.C. Eklundh, "A Fast Computer Method for Matrix Transposing," IEEE Transactions on Computers, July, 1972.
7. W. K. Pratt, "Generalized Wiener Filtering Computation Techniques," IEEE Transactions on Computers, July, 1972.
8. W.K. Pratt, "Linear and Nonlinear Filtering in the Walsh Domain," Proceedings of Applications of Walsh Functions, Washington, D.C., 1971, pp. 38-42.

ALL YOU ALWAYS WANTED TO KNOW ABOUT ELECTROMAGNETIC WALSH WAVES

Henning F. Harmuth*
Department of Electrical Engineering
The Catholic University of America
Washington, D.C. 20017

Abstract

Electromagnetic waves used in communications are predominantly sinusoidal waves. This should not lead one to believe that sinusoidal waves are the only possible or useful ones. It is shown how an optical spectrometer would have to be constructed to decompose light into Walsh functions, that such a spectrometer would yield more information than the usual spectrometer showing the frequency power spectrum, but that extremely fast time variable shutters would be needed for its implementation. Electromagnetic Walsh waves of interest in communications, on the other hand, require switches with transient times of about 1 nanosecond, which is within the state of the art. Antennas, generated field strengths, quadrupole radiation, an interference effect and the Doppler effect are discussed in a nonmathematical way.

INTRODUCTION

It is frequently believed that an electromagnetic wave must have an electric and magnetic field strength that vary with time like a sinusoidal function. Actually the generation of sinusoidal waves is a considerable technological feat. Heinrich Hertz never succeeded in producing anything close to a sinusoidal wave. His experiments with wave propagation were done with what we would call colored noise today. The generation of reasonably sinusoidal waves was a vexing problem for some twenty years following Hertz's experiments, and it was not solved satisfactorily until the invention of the electronic tube. Anybody working with fast switching circuits knows that the problem is not how to produce nonsinusoidal waves but how not to produce them. Indeed we cannot switch an electric lamp on or off without generating nonsinusoidal waves.

Visible light is a form of electromagnetic waves for which the sinusoidal time variation is often believed to have been proved by interference experiments. This is not so. Let us investigate in some detail what interference experiments prove.

* The author wants to thank the International Telephone and Telegraph Corporation, Electro-Physics Laboratories for the financial support of his work.

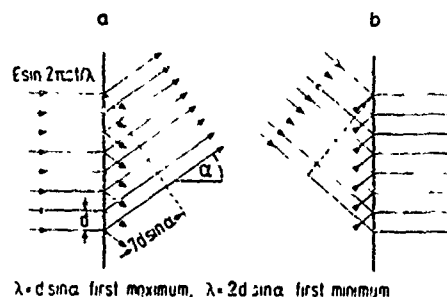


Figure 1. Linear, time invariant diffraction grating with fixed light source and movable detector (a), and with fixed detector and movable light source (b).

Figure 1a shows a diffraction grating with eight transparent slots at a distance d . A sinusoidal wave with wavelength λ coming from the left produces eight spherical waves on the right side of the grating. The waves add up for the angle $\alpha = 0$, they cancel to yield a first minimum for $\sin \alpha = \lambda/2d$, they add again to yield the first maximum for $\sin \alpha = \lambda/d$, and so forth.

If a sum of sinusoidal waves $\sum E_i \sin(2\pi c t / \lambda_i + \beta_i)$ is received from the left one obtains minima and maxima for each wave. This means that the incident light signal is decomposed into sinusoidal functions. Since we know from Fourier analysis that almost any signal can be decomposed into sinusoidal functions, the pattern of minima and maxima produced by the diffraction grating will prove to us that this device has the necessary features to actually perform such a decomposition. In other words, the diffraction pattern proves that the diffraction grating is a linear, time invariant device. One will suspect that a time variable diffraction grating will decompose light into some other system of functions.

DECOMPOSITION OF LIGHT INTO WALSH FUNCTIONS

Figure 1b shows a modification of the diffraction grating of Figure 1a. The detector observes the light emitted vertically to the grating while the light source is moved to

provide angles of incidence from $\alpha = 0$ to $\alpha = 90^\circ$. In Figure 1a, on the other hand, the detector has to be moved while the light source is fixed.

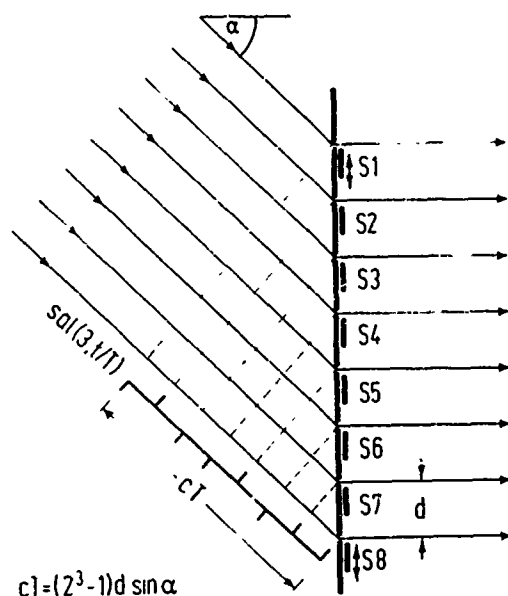


Figure 2. Linear, time variable diffraction grating with fixed detector. The time variation is provided by on-off shutters S1 to S8.

Figure 2 shows how the diffraction grating of Figure 1b has to be changed to decompose light into Walsh waves. The Walsh functions have the parameters time base T and sequence ϕ which are significant for our purpose, while only the frequency or wavelength was important for sinusoidal functions. The time base is determined by choosing the angle of incidence α . If the diffraction grating has eight slots with distance d one has to choose α according to the formula

$$cT = 7ds \sin \alpha, \quad (1)$$

where c is the velocity of light. In the general case of 2^n slots one has to substitute $2^n - 1$ for 7.

The Walsh functions $\text{sal}(i, t/T)$ and $\text{cal}(i, t/T)$ with time base T but various normalized sequences i are separated by making the transparent slots of the diffraction grating time variable. In the case of the Walsh functions this time variation is provided by on-off shutters S1 to S8 that are either open or closed. For other systems of functions one would generally need a more complicated form of time variation. Let us ignore for the moment how such shutters

could be implemented and let us find the rule for their operation.

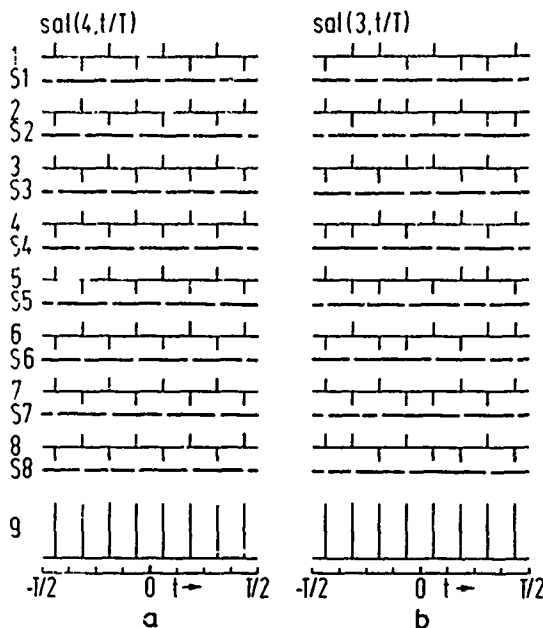


Figure 3. Operation of the shutters S1 to S8 for passage of the waves $\text{sal}(4, t/T)$ and $\text{sal}(3, t/T)$; black: shutter closed, white: shutter open. The amplitude samples shown by a heavy line illustrate the delay between the eight waves.

Figure 3 shows the operation of the shutters S1 to S8 to permit a Walsh wave $\text{sal}(4, t/T)$ to pass. The lines 1 to 8 show the function represented by samples with time shifts $0, T/8, 2T/8, \dots, 7T/8$. These time shifts correspond to the arrival of such a wave at the eight transparent slots in Figure 2. The shutters S1 to S8 are open (white) or closed (black) as shown. One may see that there are always four positive samples of the Walsh wave that are superimposed by the properly opened shutters while the negative samples are blocked. One may readily see that the opening of the shutter S1 coincides with the four positive samples of $\text{sal}(4, t/T)$ in Figure 3a and with those of $\text{sal}(3, t/T)$ in Figure 3b. The opening times of the shutters S2 to S8 are obtained from a cyclical shift of the opening times of the shutter S1.

Figure 4a shows the shutters operated for passage of $\text{sal}(3, t/T)$ but the function $\text{sal}(4, t/T)$ is applied. Two positive and two negative samples are passed through the diffraction grating at any time and their sum yields zero. Figure 4b shows the shutters

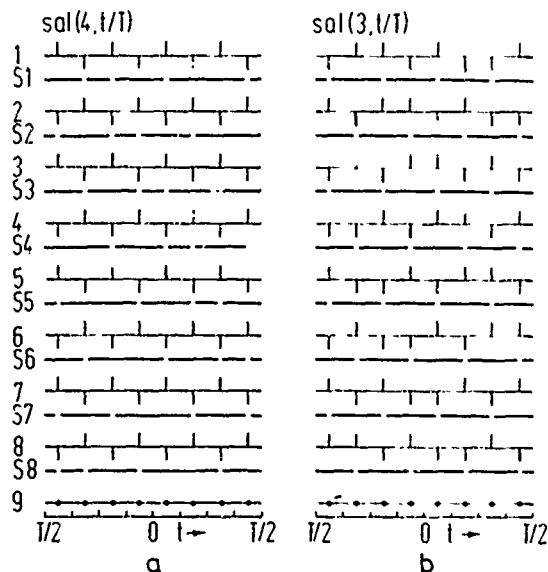


Figure 4. (a) Shutters S1 to S8 operating to pass $sal(3, t/T)$ block $sal(4, t/T)$. (b) Shutters operating to pass $sal(4, t/T)$ block $sal(3, t/T)$.

operated for passage of $sal(4, t/T)$ but $sal(3, t/T)$ is applied. Again the waves cancel at all times.

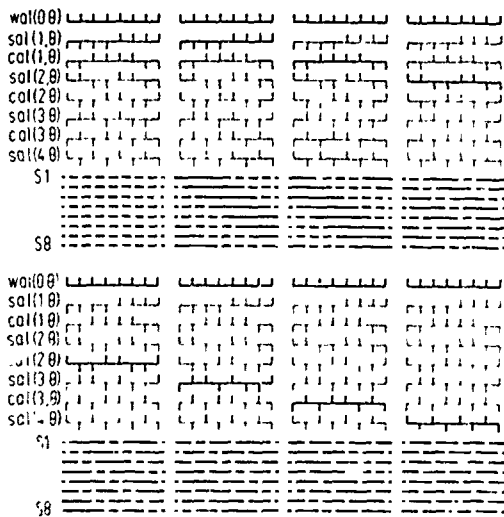


Figure 5. Operation of the shutters S1 to S8 in Figure 2 for separation of the first eight Walsh functions.

Figure 5 shows the operation of the shutters S1 to S8 for separation of the first eight Walsh functions. The functions shown

in heavy line are passed all the others (cancel by interference). Note that the constant function $wal(0,0) - wal(0, t/T)$ is always passed in addition to the desired function. A short reflection of Figure 1 shows that the same holds true for a time invariant diffraction grating - but there is no light wave with frequency or sequency zero

To obtain an estimate for the operating times of the shutters let us reflect that the frequency of visible light lies between 4×10^{14} and 8×10^{14} Hz. This calls for switching times in the order of 10^{-16} s or less to permit the use of the sampled amplitude representation of Walsh functions as shown in Figures 3 to 5. One cannot rule out that this will eventually be possible but presently known effects for the implementation of shutters, such as the Kerr effect, are some 5 to 6 orders of magnitude slower. An obvious advantage of time variable spectrometers would be that they would not only show the relative power of different spectral lines but also uncover a relationship between the times of emission.

WALSH SHAPED RADIO WAVES

Having shown that electromagnetic waves are not necessarily sinusoidal the engineer is faced with the task of showing that something useful can be done with nonsinusoidal waves. So far the investigation of nonsinusoidal electromagnetic waves has been restricted essentially to Walsh waves. Four basic differences between Walsh and sinusoidal functions have been found that might be turned into useful applications:

- The technology for the implementation of equipment is different.
- The differentiation of a sinusoidal function yields the same function except for a changed amplitude and phase, while the differentiation of a Walsh function yields a differently shaped function.
- The sum of several sinusoidal functions with arbitrary amplitudes and phases but equal frequency yields a sinusoidal function with the same frequency. This is the basis for interference effects. Walsh functions are summed differently and their interference effects are thus different.
- The Doppler effect can transform a sinusoidal function into another for any velocity ratio v/c while a ratio $v/c = 3/5$ or more is required to transform a Walsh function into another one of the same system.

Let us look at examples of these effects. Figure 6 shows a radiator for Walsh waves implemented by four Hertzian magnetic dipoles. Currents will flow clockwise in the

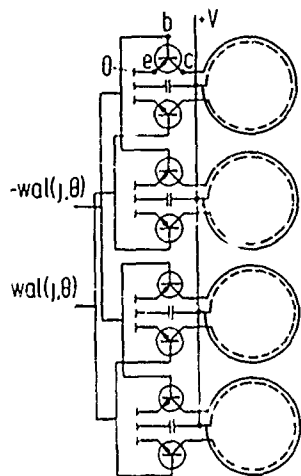


Figure 6. Array of four Hertzian magnetic dipoles.

loops shown by dashed lines if the function $wal(j, \theta)$ has a positive value. A negative value of $wal(j, \theta)$ will cause currents to flow counter clockwise in the loops shown by dashed lines. The diameter of the loops is small compared with cT , where T is the time base of the Walsh function $wal(j, \theta) = wal(j, t/T)$.

The time variation of the electric and magnetic field strengths produced by a Hertzian magnetic dipole are shown in Figure 7. The first line shows the idealized current $i(t) = I \text{cal}(3, t/T)$ flowing in the dipole. $E(1/r^2, t)$ is a component of the electric field strength declining proportionate to $1/r^2$, which is the near zone component. The far zone component declining proportionate to $1/r$ is represented by $E(1/r, t)$. The magnetic field strength consists of the three components $H(1/r^3, t)$, $H(1/r^2, t)$ and $H(1/r, t)$ which decline proportionate to $1/r^3$, $1/r^2$ and $1/r$.

The time variation of the far zone components $E(1/r, t)$ and $H(1/r, t)$ is the first derivative of the dipole current $i(t)$. The components $E(1/r^2, t)$ and $H(1/r^2, t)$ vary like the current $i(t)$, and the component $H(1/r^3, t)$ varies like the integral of the current $i(t)$. These relations between the time variation of the dipole current $i(t)$ and the components of the electric and magnetic field strengths hold true for any current $i(t)$. If $i(t)$ varies like a sinusoidal function its derivative and its integral will vary like $i(t)$ except for phase shifts. Hence, the near and far zone components of sinusoidal waves are hard to separate while those of Walsh waves can be much more readily

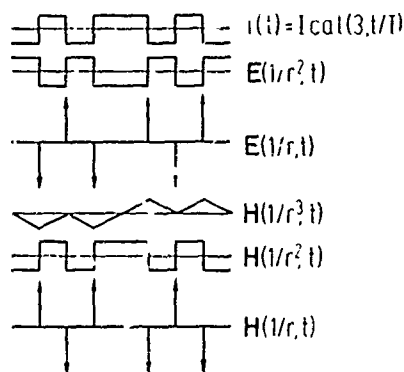


Figure 7. Time variation of the antenna current $i(t)$ in a Hertzian magnetic dipole and the time variations of the produced electric and magnetic field strengths declining proportionate to $1/r$, $1/r^2$ and $1/r^3$.

separated due to their different shape. If one can separate $E(1/r, t)$ and $E(1/r^2, t)$ one may compare their power and derive the distance of the receiver. One interesting aspect of this passive distance measuring effect is that one might be able to obtain a signal-to-noise power ratio that decreases like $1/r^3$ while the active distance measurement by the radar principle yields a signal-to-noise power ratio that decreases like $1/r^4$. However, the radar principle permits a much higher accuracy of distance measurement.

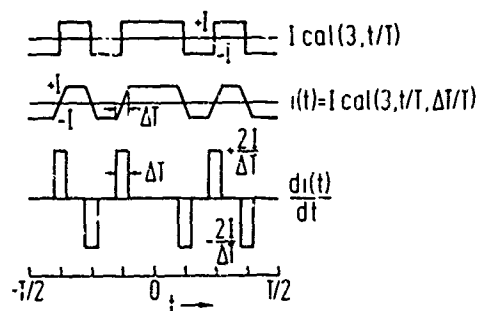


Figure 8. Walsh shaped antenna current $i(t)$ with a finite rise time ΔT and its first derivative $di(t)/dt$.

Figure 8 shows the idealized antenna current $I \text{cal}(3, t/T)$ replaced by a more realistic current $i(t)$ with a switching time ΔT from -1 to -1 or from -1 to -1 . The first derivative consists of rectangular pulses of duration ΔT and magnitude $2I/\Delta T$. When ΔT approaches zero one obtains the Dirac

pulses of Figure 7. The energy of a rectangular pulse is proportionate to $(2I/\Delta T)^2 \Delta T = 4I^2/\Delta T$. Hence, the average power of the electric and magnetic field strength in the far zone cannot only be increased by a larger amplitude I of the antenna current but also by a shorter switching time ΔT . This indicates that a small antenna can radiate a large average power.

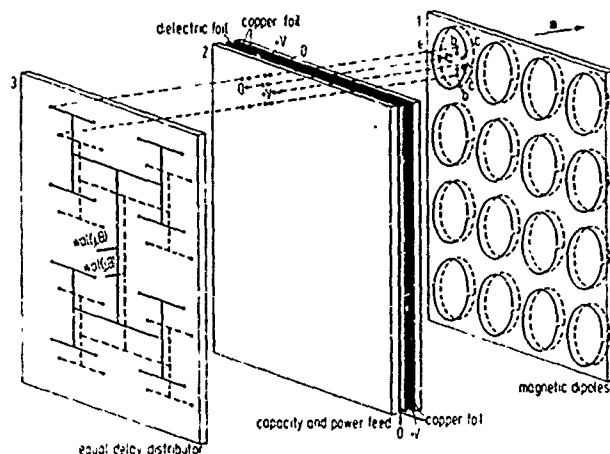


Figure 9. Two-dimensional array of 16 Hertzian magnetic dipoles.

Figure 9 shows an array of 16 Hertzian magnetic dipoles. They are arranged in a two-dimensional pattern. Let us observe that the power radiated by n interacting dipoles is n^2 times the power radiated by one dipole and that the power radiated by one dipole increases proportionate to n . There is thus a great incentive to use many interacting dipoles. To make the dipoles interacting they must not be spaced too far apart; more precisely, if the distance between two dipoles is d it will take the time d/c for a change of current in one dipole to affect the current flowing in the other. Obviously we can have more Hertzian dipoles close together if they are arranged in a two-dimensional array rather than in a one-dimensional array as in Figure 6. Even more desirable is an extension of the radiator of Figure 9 to a three-dimensional array.

Antennas for sinusoidal waves are usually one-dimensional. The typical resonant dipole is long in one space dimension and has virtually no extension in the two other space dimensions. One could of course use the antennas of Figures 6 and 9 to radiate sinusoidal waves but the generators delivering the currents to the Hertzian magnetic dipoles could no longer be simple switches but would have to be amplifiers for sinusoidal currents.

Consider a transmitting antenna consisting of a parabolic reflector and a radiator at its

focal point. A two-dimensional array according to Figure 9 and even more so its extension to a three-dimensional array will come closer to a point-like radiator than the one-dimensional array of Figure 6 or the one-dimensional resonant dipole used for sinusoidal radiation. This is a second incentive for the development of two- and three-dimensional radiators.

QUADRUPOLE RADIATION

Sinusoidal waves in free space are generally dipole waves. The main reason seems to be that quadrupole and higher order multipole waves are radiated with less power by a radiator of fixed size. This is not so for Walsh waves.

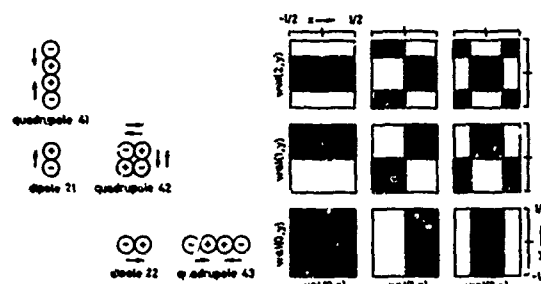


Figure 10. Classification of dipoles, quadrupoles and multipoles by two-dimensional Walsh functions $wal(k, x)wal(m, y)$; $k, m = 0, 1, 2$.

Figure 10 shows on the left vertically and horizontally polarized dipoles denoted "dipole 21" and "dipole 22". Furthermore, there are three quadrupoles denoted 41 to 43. The right side of Figure 10 shows two-dimensional Walsh functions. The black areas represent the value +1, the white areas the value -1. One may readily see that the positive and negative signs of the dipoles on the left correspond to the positive and negative signs of the two-dimensional Walsh functions. This correspondence greatly simplifies a discussion of radiation modes. For instance, it is obvious from Figure 10 that there is no "unipole" corresponding to the all black function $wal(0, x)wal(0, y)$. Unipole radiation does not exist for electromagnetic waves due to the preservation of charge but it is the major mode of radiation of acoustic waves.

According to Figure 10 a quadrupole 41 consists of two dipoles fed with currents flowing in the opposite direction. Two such electric Hertzian dipoles are shown in detail in Figure 11. The electric and magnetic field strengths in the far zone, declining proportionate to $1/r$, vary with time like the second derivative of the current flowing in

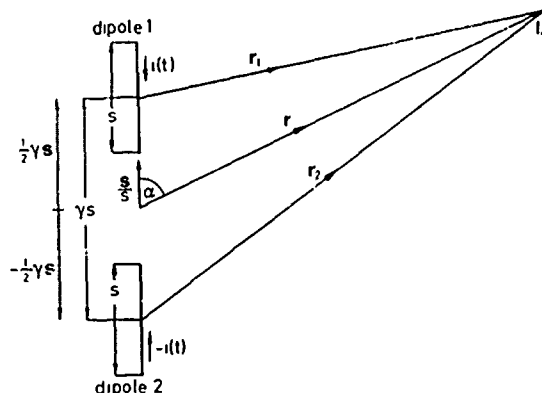


Figure 11. Hertzian electric quadrupole 41.

the dipoles. The resulting time variation is shown in Figure 12. On top is the nominal dipole current $I \cos(3\pi t/T)$. The second line shows the more realistic current with finite transient time between $+I$ and $-I$; furthermore, the transients are rounded while those in Figure 8 had sharp breaks. As a result the first derivative shown in line 3 consists of trapezoidal rather than rectangular pulses of width ΔT . The second derivative consists of pairs of rectangular pulses of width $\epsilon \Delta T$. The energy of a rectangular pulse is proportionate to

$$\left[2I / \epsilon (\Delta T)^2 \right]^2 \Delta T = 4I^2 / \epsilon^2 (\Delta T)^3. \text{ Hence,}$$

the average power of the electric and magnetic field strength in the far zone increases faster with decreasing switching time than for dipole radiation. This means that it is theoretically possible to radiate more power in the quadrupole mode than in the dipole mode for an antenna of given size if one succeeds in decreasing the switching time sufficiently. The different time variation of the electric and magnetic field strengths for dipole and quadrupole radiation indicates that interference effects will be different. Generally speaking, interference effects of quadrupole radiation yield a better resolution than interference effects of dipole radiation, and this holds even more so for higher order multipole radiation.

INTERFERENCE EFFECTS, DOPPLER EFFECT

Figure 13 shows as an example of an interference effect the reflection of a sinusoidal radar pulse from two point-like targets B1 and B2. The reflected pulses are shown in lines a and b. Line c shows their sum, which is the signal received by the radar. Except for the bulges of duration $2(d_2 - d_1)/c$ at the beginning and end of the signal there is nothing that indicates that this signal was

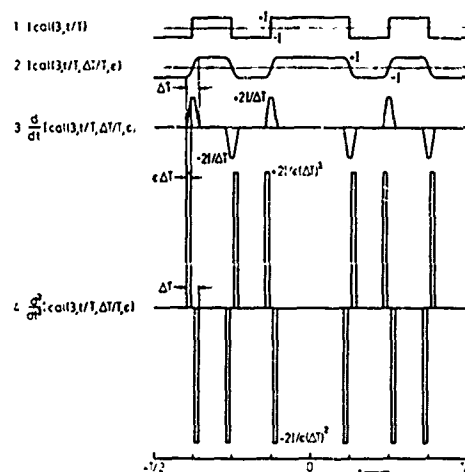


Figure 12. Time plots for quadrupole radiation. (1) nominal antenna current; (2) realistic antenna current; (3) first derivative of the antenna current; (4) second derivative of the antenna current or time variation of the electric and magnetic field strength in the far zone.

caused by two targets rather than by a single larger target. A typical radar pulse will contain about 1000 carrier cycles, hence the relative energy of the bulges at the end is very small.

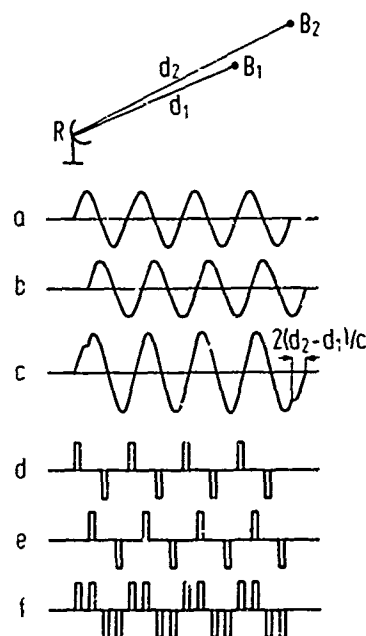


Figure 13. Example of an interference effect of sinusoidal and Walsh shaped waves.

Let a Walsh wave in dipole mode be reflected by the two targets. The far zone components of the electric and magnetic field strengths consist of narrow rectangular pulses. The pulses reflected from B1 are shown in line d and those reflected from B2 are shown in line f. The difference between a reflection from two small targets or one larger target is no longer represented by low energy effects at beginning and end of a radar pulse. Indeed, even a periodic Walsh wave would distinguish between one and two targets.

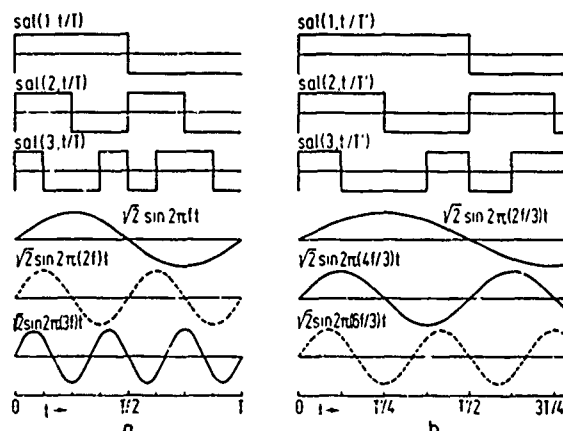


Figure 14. Doppler effect for sinusoidal and Walsh shaped waves: (a) functions generated and observed in the same system of reference; (b) functions observed in a system moving away with velocity $-v/c = 5/13$.

Figure 14a shows three Walsh and three sine functions as observed in a system of reference that has no relative velocity to the system of reference in which they are generated. Figure 14b shows the observed functions if the system in which they are generated moves away with velocity $-v/c = 5/13$. An observer cannot tell whether the observed sine function $\sin 2\pi(6f/3)t$ was produced by a transmitter without relative velocity radiating the function $\sin 2\pi(2f)t$ or one with relative velocity $-v/c = 5/13$ radiating the function $\sin 2\pi(3f)t$. In the case of the Walsh function there is no difficulty telling that $\text{sal}(3, t/T')$ in Figure 14b was caused by $\text{sal}(3, t/T)$ in Figure 14a and not by $\text{sal}(2, t/T)$. The same applies to the

first or second derivatives of the Walsh functions.

If transmitter and receiver move away from each other the relative velocity must be at least $-v/c = 3/5$, if they approach each other it must be at least $+v/c = 3/5$ before a Walsh function with known time base T is transformed into another. There are no such minimum velocities for sinusoidal functions.

REFERENCES

1. C. Huygens, *Treatise on Light* (1690). Dover: New York 1962. Huygens introduced the concept of light waves but he did not assume that they have a sinusoidal time variation.
2. J. Pearlman, Radiation patterns for antennas with Walsh current input. *Proc. 1970 Walsh Functions Symposium*, 65-69, AD 707-431.
3. H. Harmuth, Electromagnetic Walsh waves in communications. *Proc. 1970 Walsh Functions Symposium*, 248-259, AD 707-431.
4. T. Frank, Circuitry for the reception of Walsh waves. *Proc. 1971 Symposium on Theory and Application of Walsh Functions*. The Hatfield Polytechnic, Hatfield, Hertfordshire, England.
5. H. Harmuth and T. Frank, Theoretical study of the synchronous demodulation problem for a Walsh wave receiver. AD 725-755.
6. H. Harmuth, *Transmission of Information by Orthogonal Functions*, 2nd edition. Springer-Verlag: New York 1972.
7. J. E. Gibbs and H. A. Gebbie, Application of Walsh functions to transform spectroscopy. *Nature*, vol. 224 (6 Dec. 1969), 1012-1013.
8. J. A. Decker, Experimental realization of the multiplex advantage with a Hadamard spectrometer. *Applied Optics*, vol. 10 (1971), 510-514.

WALSH FUNCTIONS IN GRILLE SPECTROSCOPY

by

Alvin M. Despain
Utah State University

George A. Vanasse
Air Force Cambridge Research Laboratories

Abstract

Walsh Functions have been very useful in transform spectroscopy as evidenced by the recent work in Hadamard transform spectroscopy. It is also possible to use Walsh functions to generate a suitable grille for use in grille spectroscopy as developed by Girard. The hyperbolic grill that is often used in this technique is very closely related to the grill whose transmittance is defined by a complete set of the sal functions. This paper explores this relationship and indicates how other grilles of specified properties may be generated. Thus, the application of Walsh functions can be profitably extended to the domain of the grille spectrometer.

Introduction

The purpose of this paper is to explore how Walsh functions may be effectively employed to greatly improve the performance of grating spectrometers by using Walsh functions patterned grilles (masks) to increase the light throughput and to multiplex the various wavelengths to be examined. A typical grating spectrometer is illustrated in Figure 1 and may be schematically diagramed as shown in Figure 2. The function of the grating is to produce an angular deviation of the light beam impinging upon it in proportion to the wavelength of the light.

The original light beam is defined by the entrance mask and collimated by the mirror before falling on the grating. The angle at which the light is reflected from the grating is a function of wavelength. This beam is focused by the mirror onto the output mask. A detector behind this mask then converts the light to an electrical signal for recording.

The sensitivity of the spectrometer is a measure of the weakest light signal it can detect and is a function of the effective area and solid angle over which light may be gathered (throughput) and the noise level of the detector employed. The throughput E_g of a conventional spectrometer is proportional to several instrument parameters.

$$E_g \propto A_g \lambda / R$$

where

A_g = grating area

R = resolving power

λ = "λ" number = L/F

L = entrance aperture height

F = focal length of mirror

and it is assumed that the system is "energy" limited not diffraction limited. The resolution in the usual spectrometer is generally inversely proportional to the entrance aperture (slit) width.

A grille or mask may be employed as an entrance aperture, exit aperture, or both, to improve the performance of the system. These grilles were first employed by Golay (Golay, 1949; Girard, 1963) to increase the effective entrance and exit apertures without sacrificing resolution (throughput advantage). It is also possible to use them to multiplex various wavelengths using one detector, (Golay, 1949) thus improving the signal-to-noise ratio of the measurement compared with that obtained by recording each wavelength in sequence (multiplex advantage). A grille spectrometer may have both the throughput advantage and the multiplex advantage (Harwit, et al., 1970) and can compete with an interferometric spectrometer, without requiring the extreme mechanical stability of the interferometer (Vanasse and Sakai, 1967). A great variety of grilles have been employed as summarized in Table I. Golay (1953), the pioneer of this field, mentioned that Walsh functions might be useful as grille patterns, but they have only been recently employed despite their natural advantage of being a complete orthonormal set of "on-off" form.

General Formulation

The operation of a grille spectrometer may be mathematically modeled by describing the total radiation intensity incident on the detector in terms of the incident radiation

spectrum and the functions that describe each element of the entrance and exit grilles (see Figure 2). Each element is a slot whose width determines the spectral resolution of the spectrometer, just as the single slit does for a monochromator. The entire entrance and exit apertures are filled with slots and each slot can, in general, be modulated (opened or closed) independently of any other slot. As shown in Figure 2, the effect of the grating is to cause radiation of amplitude λ_i and wavelength number i entering slot j to be shifted to the exit slot $i + j$.

In general, we wish to encode each slot in the spectrometer so that we can measure and record all the exiting radiation and then recover an estimate of the spectral elements $\{\lambda_i\}$. Thus, let ψ_j represent the encoding function for entrance slot j . Similarly ϕ_k will represent the exit slot encoding function. The total exiting radiation is therefore described by summing all radiation over all wavelengths that enter each entrance slot and leave each exit slot. Thus,

$$S = \sum_i \sum_j \sum_k \lambda_i \psi_j \phi_k \delta(i + j - k)$$

where S is the signal that is detected, recorded and later analyzed to recover $\{\lambda_i\}$ and δ is the unit delta function. The function ψ and ϕ can of course be, in general, any of a very large set of functions. However, our interest here is to restrict each ψ and ϕ to be some modified combination of Walsh functions; modifying referring to the on-off properties. Thus, we define an offset Walsh function X_j

$$X_j(\theta) = \frac{\text{Wal}(j, \theta) + 1}{2},$$

where $\text{Wal}(j, \theta)$ is the usual notation for the Walsh functions. The functions ϕ and ψ are often combinations of the X 's which take on only the values 0 or 1.

It is a well known property of Walsh functions (Harmuth, 1970; p.20) that

$$\text{Wal}(h, \theta) \text{Wal}(k, \theta) = \text{Wal}(h \oplus k, \theta),$$

where the operator \oplus signifies the module 2 sum. Thus, it is found that

$$2X_j X_k = X_{j \oplus k} + X_j + X_k - 1.$$

For the case where ψ and ϕ are taken as "offset Walsh functions" then

$$2S(\theta) = \sum_i \sum_j \sum_k \lambda_i \delta[f(i, j, k)]$$

$$\cdot [X_{j \oplus k}(\theta) + X_j(\theta) + X_k(\theta)]$$

where the $f(i, j, k)$ determines how the X functions are assigned to the slots. Since the

Walsh and hence the X functions form an orthogonal set, each of the above terms (that are unique) can be extracted from the recorded signal S by Walsh-Fourier analysis.

Multiplex Advantage Grille Spectrometers

The Grille Spectrometers of Golay (1951), Ibbett *et al.* (1968), Sloane *et al.* (1969), may be described by choosing the constants one for a single entrance slot and zero for the remaining entrance slots. Then orthogonal functions are chosen for the exit slots, thus multiplexing the amplitudes of the spectral elements onto a single signal. An excellent choice for the exit slot functions is, of course, the "offset Walsh functions" (see Figure 3 for an example) and the radiation spectrum may be recovered by a Walsh-Fourier transformation. Thus,

$$2S(\theta) = \sum_i \sum_j \sum_k \lambda_i \delta(j) \delta(i + j - k)$$

$$\cdot [X_{j \oplus k}(\theta) + X_j(\theta) + X_k(\theta) - 1]$$

$$S(\theta) = \sum_i \lambda_i X_i(\theta)$$

and thus the spectrum is easily recovered as

$$\hat{\lambda} = \text{Walsh Transform } [S(\theta)]$$

Throughput Advantage Grille Spectrometers

The Grille Spectrometers of Golay (1951), Girard (1963), and others employ identical (except for displacement) entrance and exit grilles but record only the average value of the modulated signal and exhibit very high luminosity. In analogy with the Girard grille, a grille herein named a sal grille may be formed using the "offset sal functions". This grille may be compared with Girard's grille in Figure 3. The output signal is given by the average of

$$2S(\theta) = \sum_i \sum_j \sum_k \lambda_i \delta(i + 2j - 2k) \delta(2k - n)$$

$$\cdot [X_{2j \oplus 2k}(\theta) + X_{2j}(\theta) + X_{2k}(\theta) - 1]$$

or

$$2S(\theta) = \sum_i \sum_j \lambda_i [X_{(n-i) \oplus n}(\theta) +$$

$$X_{n-i}(\theta) + X_n(\theta) - 1]$$

where n = displaced number of slots as the exit mask is moved across the exit aperture. Since $X_n = 0$ if $n \neq 0$, then for $n \neq 0$

$$2S(\theta) = \sum_i \sum_{j=1}^{2j} \delta(n-i) [X_0(\theta) + X_0(\theta) + 0 - 1]$$

$$S(\theta) = J \sum_i \lambda_i \delta(n-i) + J$$

Thus, for position n , the signal averaged over the independent variable θ , produces a measure of the spectral component λ_n . Note the throughput advantage indicated by the factor J .

It is interesting to note that in practice, difficulties were experienced with the Girard grille in that nonuniform illumination on the entrance aperture produced serious distortions. Moret-Bailly *et al.* (1970) developed a "random grille" composed of slots "randomly" chosen to be open or closed to overcome this problem. A similar grille can be realized using a pseudo-random sequence which may in some cases be a certain Walsh function with the leading element deleted. These sequences would be pseudo-randomly arranged to produce a pseudo-random grille with properties very similar to the "random" grille of Moret-Bailly *et al.*

Multiplex and Throughput Advantage Grille Spectrometers

The grille spectrometer of Harwit *et al.* (1970) employed grille functions derived from a Hadamard matrix which for some cases (but not most) are also Walsh functions. Separate masks are required for the entrance and exit apertures and they must be displaced independently. Nevertheless, this configuration with both the throughput and multiplex advantage greatly improves the performance of a spectrometer.

It is also possible to derive a grille spectrometer with the above properties but with the important advantage that a single common mask can be used for both the entrance and exit apertures. Such an arrangement is shown in Figure 4 for a Littrow mount. This "Walsh grille spectrometer" would be very similar to the mock interferometer devised by Mertz (1965), but would be without its disadvantages and would employ "offset Walsh functions" for encoding purposes. Thus, in terms of the above formulations let $\psi_i = \phi_k = X_j$, then

$$2S = \sum_i \sum_j \lambda_i X(i+j)\psi_j + X_{i+j} + X_j - 1]$$

$$2S = \tilde{X}^t \tilde{C}^t \tilde{X}$$

Now by a Walsh-Fourier transformation of S the sequency amplitudes \tilde{S} can be recovered. Thus

$$\tilde{C}^t \tilde{X} = 2\tilde{S}$$

In general \tilde{C} is not square. Therefore, the least square estimate of the amplitudes of the spectrum \tilde{X} is

$$\tilde{C}^t \tilde{C}^t \tilde{X} = 2\tilde{C}^t \tilde{S}$$

$$\tilde{X} = 2[\tilde{C}^t \tilde{C}^t]^{-1} \tilde{C}^t \tilde{S} = \tilde{D} \tilde{S}$$

where

$$\tilde{D} = 2[\tilde{C}^t \tilde{C}^t]^{-1} \tilde{C}^t$$

An example of how the spectrum \tilde{X} can be recovered for the case where four separate elements are to be ascertained illustrates the design of such an instrument. It is convenient to use a mask of $2n - 1$ or seven slots for $n = 4$ spectral elements (this is not a requirement, however). Consequently,

$$2S = \sum_{i=0}^3 \sum_{j=1}^7 \lambda_i [X_{(i+j)}\psi_j + X_{i+j} + X_j - 1]$$

can be expanded and common terms collected to form $2S = \tilde{X}^t \tilde{C}^t \tilde{X}$ where

4	-4	-4	-4
1	6	0	0
1	1	6	0
1	2	1	5
1	1	1	1
0	1	1	3
0	0	3	1
0	1	0	2

$\tilde{C}^t =$

\tilde{D} is then found

$$\tilde{D} = 2[\tilde{C}^t \tilde{C}^t]^{-1} \tilde{C}^t$$

or $\tilde{D}^t =$

.176144	-.050292	-.022536	-.009004
.029118	.150659	-.026963	-.063176
.075509	-.005634	.125660	-.046127
.100225	-.014821	-.021023	.126166
.066880	.003793	.012464	.013143
.024363	-.015691	-.005163	.070177
.018691	-.027126	.056548	.008422
.016453	.001309	-.019275	.047543

Now to estimate the spectrum of the radiation that impinges on the entrance aperture, the coded disk is rotated and the signal S is recorded. A Walsh-Fourier transform of S yields the sequency components \tilde{S} . Then the spectrum $\tilde{X} = \tilde{D} \tilde{S}$.

Summary

Grille spectrometers have a great advantage over the usual monochromator and compare favorably with the interferometric spectrometers in sensitivity. Grille patterns of offset Walsh functions produce especially useful instruments. Three new configurations of grille spectrometers, the sal grille, the two-dimensional pseudo-random grille, and the Walsh grille

spectrometers were synthesized using Walsh functions. The Walsh grille spectrometers has several significant advantages over all the other types of grille spectrometers, exhibiting both the throughput and multiplex advantage using a single common mask for both the entrance and exit apertures and requiring no more than $2n - 1$ measurements if n spectral elements are to be determined.

Table 1. GRILLE PATTERNS FOR GRILLE SPECTROMETERS

GRILLE DESCRIPTION	REFERENCE	EMPLOY WALSH FUNCTIONS	THROUGHPUT ADVANTAGE	MULTIPLEX ADVANTAGE
Complementary series	Golay (1949)	No	Yes	No
Hyperbolic	Girard (1963)	No	Yes	No
Mock	Hertz (1965)	No	Yes	Yes
Circular symmetry	Tinsley (1966)	No	Yes	No
Fourier coded	Grainger <i>et al.</i> (1967)	No	No	Yes
Truncated Walsh	Ibbett <i>et al.</i> (1968)	Yes	No	Yes
Algebraic code	Decker and Harwitt (1968)	No	No	Yes
Hadamard - single encoded	Sloane <i>et al.</i> (1969)	No	No	Yes
Hyperbolic	Girard (1970)	No	No	Yes
Pseudo-random - one dimensional	deGraaw and Veltman (1970)	No	No	Yes
Hadamard - double encoded	Harwitt <i>et al.</i> (1970)	Yes	Yes	Yes
Random	Moret-Bailly <i>et al.</i> (1970)	No	Yes	No
Sal function	Despain and Vanasse (1972)*	Yes	Yes	No
Pseudo random - two dimensional	Despain and Vanasse (1972)*	Yes	Yes	No
Walsh function	Despain and Vanasse (1972)*	Yes	Yes	Yes

*These grilles are described in this paper.

References

1. Decker, John A., Jr., and Martin O. Harwitt, "Sequential Encoding with Multislit Spectrometers," *Appl. Opt.*, **7**, 2205; 1968.
2. deGraaw, Th., and B.P. Th. Veltman, "Pseudo-Random Binary Sequences for Multiplex Codes," *Appl. Opt.*, **9**, 2658; 1970.
3. Girard, A., "Spectromètre à Grille," *Appl. Opt.*, **2**, 79; 1963.
4. Girard, A., "Devices for Multiplex Stellar Spectroscopy," *Proceedings of the Aspen International Conference On Fourier Spectroscopy*, George A. Vanasse, *et al.*, Ed., (Air Force Cambridge Research Laboratories, Special Report, No. 114), p.425; 1971.
5. Golay, Marcel J.E., "Multi-Slit Spectrometry," *JOSA*, **39**, 437; 1949.
6. Golay, Marcel J.E., "Static Multislit Spectrometry and its Application To The Panoramic Display of Infrared Spectra," *JOSA*, **41**, 468; 1951.
7. Golay, Marcel J.E., "Multislit Multipass Spectrometers," *JOSA*, **43**, 58; 1953.
8. Grainger, J.F., J. Ring, and J.H. Stell, *J. Phys.* **28**, Colloque C2, Suppl. 3-4, 44; 1967. (as quoted by Ibbett 1968)

9. Harmuth, Henning F., "Transmission of Information by Orthogonal Functions," Springer-Verlag, New York, p.20; 1970.
10. Harwit, M., P.G. Phillips, T. Fine, and N. J.A. Sloane, "Doubly Multiplexed Dispersive Spectrometers," *Appl. Opt.*, 9, 1149; 1970.
11. Ibbett, R.N., D. Aspinall, and J.F. Grainger, "Real-Time Multiplexing of Dispersed Spectra in Any Wavelength Region," *Appl. Opt.*, 7, 1089; 1968.
12. Mertz, Lawrence, "Transformations in Optics," John Wiley and Sons, Inc., New York, p.64; 1965.
13. Moret-Bailly, J., Cl. Milan, and J. Cadot, "Un Nouveau Type de Grille de Spectrometre," *Nouvelle Revue D'Optique Appliquee*, I, p.137; 1970.
14. Sloane, N.J.A., T. Fine, P.G. Phillips, and M. Harwit, "Codes for Multiplex Spectrometry," *Appl. Opt.*, 8, 2103; 1969.
15. Tinsley, B.A., "The Circularly Symmetric Grille Spectrometer," *Appl. Opt.*, 5, 1139; 1966.
16. Vanasse, G.A., and H. Sakai, *Progress in Optics*, E. Wolf, Ed., Wiley-Interscience, New York, 7, p.298; 1967.

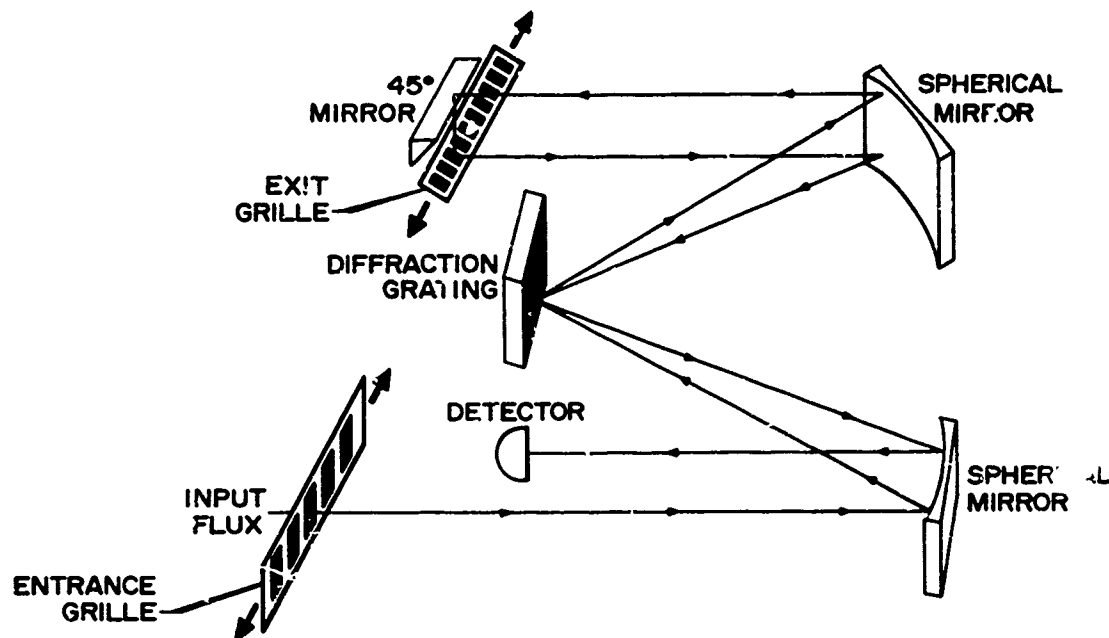
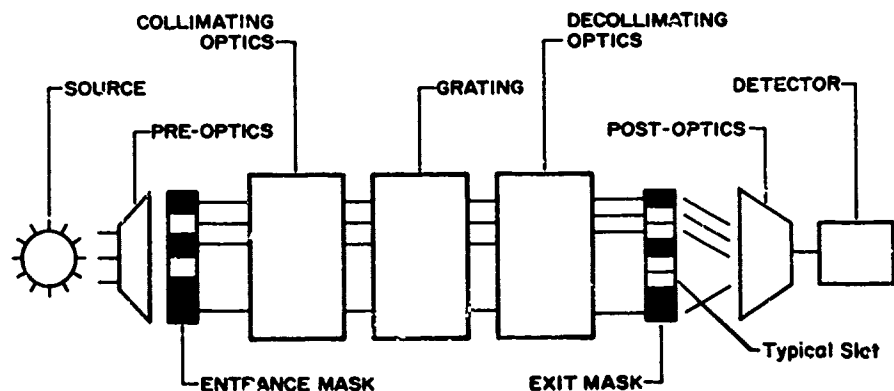


FIGURE 1 GRILLE SPECTROMETER



**FIGURE 2 SCHEMATIC DIAGRAM
GRILLE SPECTROMETER**

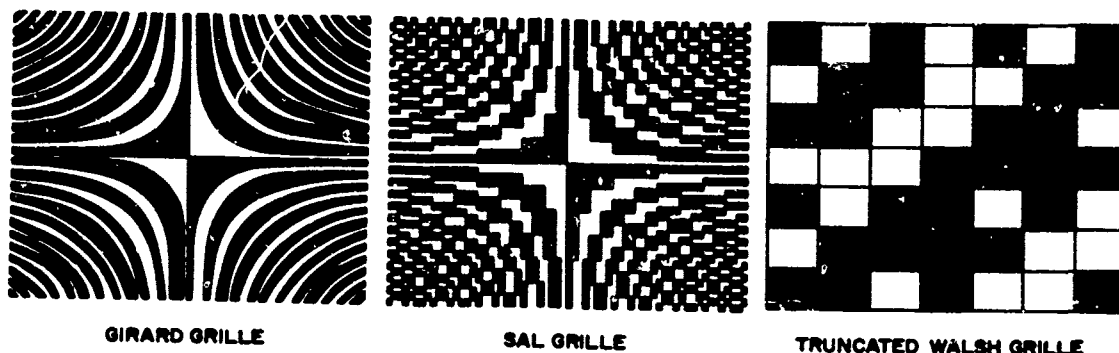


FIG.3 GRILLE PATTERNS

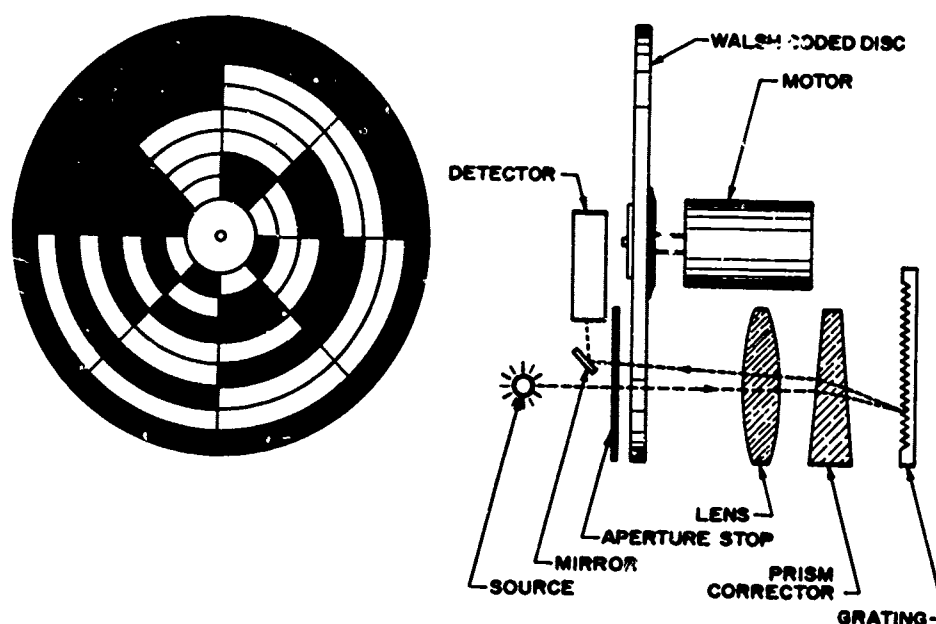


FIG.4 WALSH GRILLE SPECTROMETER

HADAMARD TRANSFORM SCINTILLATION COUNTER

by

L.M. Soroko

Joint Institute for Nuclear Research
Dubna, USSR

Abstract

The Hadamard transform scintillation counter designed to measure the angular or spatial distributions of nuclear and atomic particles in the presence of heavy noise from the photomultiplier is described. The gain in the effective signal-to-noise ratio achieved by this counter in the result of the spatial multiplexing, that is, of coding the spatial or angular distribution to be measured before counting it.

The counter consists of a multi-element heterogeneous scintillator, two lightguides, two photomultipliers and an electrical pulse bipolar counter. The multiscintillator consists of $(2M-1)$ scintillation elements. The lightflashes can enter only the lower or only the higher lightguides, the $(M-1)$ elements of the multi-scintillator being outside the lightguide.

The counting rate to be measured is equal to the difference between the number of $N(i)$ events and the number of $N(i)$ events at all M positions $(0 \leq i \leq M-1)$ of the movable multi-scintillator relative to the lightguides. It is needed to perform M independent measurements.

The spatial distribution of particles undergoes the binary coding according to the Hadamard transform algorithm. The system of the multiplex Hadamard transform detection gives rise to the gain in the signal-to-noise ratio, being equal to $(M+1)/\sqrt{M}$, where M is the number of the element-scintillators in the multiplex block scintillator. Neither the reduction of the angular resolution nor the loss of the counting rate of the useful events take place. The Hadamard transform scintillation counter gives the direct efficiency gain in the working time of particle accelerators and also in the cost of the equipment.

Introduction

The multiplex transform detectors of particles provide a new class of instruments designed for use in the nuclear physics laboratories. The basis for the existence of such instruments is the renouce of the traditional lo-

gics by sampling point by point of the unknown distribution. In contrast to this, the subject to be measured in the multiplex transform system is the coded integral transform of the function describing the distribution to be investigated.

The multiplex system of particle detection is characterized by high immunity against the different foreign perturbations. This property opens up the possibility to overcome the limit put up by the traditional logics of observation.

It was M.H. Fizeau^{/1/} who performed, more than a hundred years ago, the first multiplex measurement in optics. At the late forties the ideas of multiplex optical spectroscopy arose^{/2/}. The basis of the present paper is as follows: the multiplex communication, particularly, Walsh Hadamard multiplex systems^{/3/}, modern multiplex Hadamard transform spectroscopy^{/4,9,11/} and the correlation time-of-flight method in neutron spectroscopy^{/5,6/}. The multiplex Hadamard transform scintillation counter is described below^{/7/}.

Traditional Logics of the Experiment

In performing the experiment the investigator always must choose the optimum relation between the factors which are mutually exclusive. For example, the rise of the counting rate by increasing the sizes of the scintillator (Fig. 1) inevitably makes the spatial resolution of the counter worse. Particularly a hopeless task awaits the investigator when the useful events are so rare that the noise from the counter becomes the principal limiting factor, and the noise-error exceeds the statistical error.

It is easy to show that any ways for increasing the counting rate would come to the loss of spatial resolution. There are traditional solutions of this conflicting situation. One of them consists in the increasing of the working time of the expensive particle accelerator. Another way of the solution is the converting of the one-channel sys-

tem into the multi-channel equipment. The economical after-effects of these methods are obvious.

Multi-Element Scintillator

Meanwhile there is a nontraditional solution, namely, the use of the multiplex transform system of particle detection. The passage to the multiplex scintillation counter can be performed as follows. First of all, the spatial resolution of the system must be left unchangable. For this aim we take several element-scintillators and one photomultiplier. The group of element-scintillators is arranged in a such a way that gaps between some elements are left. The counting strips and the gaps alternate one another. The law of this alternation is given by the pseudorandom discrete sequence $g(i)$. The number M of elements must be chosen according to the cyclic transorthogonality ^{/8/}. This property is expressed by the relation

$$g(i) \times g(i) = \sum_{k=0}^{M-1} g(k) g(i+k) = (M+1) \delta(i) - 1(i) \quad (1)$$

and can be satisfied only for several values of M . For example, for $M = 11$, the function $g(i)$ is as follows:

$$+1, +1, -1, +1, +1, +1, -1, -1, +1, -1. \quad (2)$$

The function $g(i)$ can be subdivided into the two unipolar terms

$$g(i) = g_+(i) + g_-(i) \quad (3)$$

where

$$g_+(i) = +1, +1, 0, +1, +1, +1, 0, 0, 0, +1, 0 \quad (4)$$

and

$$g_-(i) = 0, 0, -1, 0, 0, 0, -1, -1, -1, 0, -1. \quad (5)$$

The unipolar system $g_+(i)$ of several elements (+1) and gaps (0) is shown in Fig. 2. The multiplex advantage of this unipolar multiplex system ^{/4/} is equal to

$$\frac{M+1}{2\sqrt{M}} \quad (6)$$

i.e. the gain in the effective signal-to-noise ratio achieved by this unipolar system. However, the unipolar system $g_+(i)$ is not yet optimal. It is desirable and effecient to take the bipolar system. For this aim we take another comb with many element-scintillators and gaps according the $g_-(i)$ function. Both system supplement each other. The multiplex advantage of such a bipolar system is equal to

$$\frac{M+1}{\sqrt{M}} \quad (7)$$

The light-flashes can enter only the lower or only the higher lightguides (Fig. 3), the $(M-1)$ elements of the scintillator being outside the two lightguides. The whole heterogene scintillator can be moved between the two lightguides.

The electrical pulses from the photomultipliers are fed to the counting device, the upper channel being in the anti-phase. Therefore, the counting rate of the upper system is subtracted from the counting rate of the lower system. The intermediate data measured by this bipolar system is the difference of the counting rates (Fig.4).

$$\Delta N(i) = N_+(i) - N_-(i) \quad (8)$$

between the number of pulses from the lower $N_+(i)$ system and the number of pulses from the upper $N_-(i)$ system. This difference must be sampled M times for different mutually independent positions of the movable scintillator. The values of $\Delta N(i)$ can be both negative and positive.

General Description of the Operation

The measurements performed by the multiplex scintillation counter are made in two stages: the detection stage and the processing one. In the course of the first stage the function $f(i)$ to be searched is converted into the integral transform

$$r(i) = f(i) \times g(i) \quad (9)$$

where \times is the operation of convolution which is performed simultaneously in the course of the measurements. The transform $r(i)$ is sampled at all values of i , from $i_{\text{int}} = 0$ till $i_{\text{final}} = M-1$. The really measured function $s(i)$ is equal to

$$s(i) = f(i) \times g(i) + \Delta n(i) \quad (10)$$

where

$$\Delta n(i) = n_1(i) - n_2(i) \quad (11)$$

and $n_k(i)$ ($k = 1, 2$) is the noise counting rate of the k -photomultiplier at all M positions of the movable multiplex scintillator. Obviously

$$\sum_{i=0}^{M-1} \Delta n(i) = M [\Delta n(i)]_{av} = 0 \quad (12)$$

The intermediate data $s(i)$ are to be retransformed according to the algorithm

$$f_{red}(i) = \frac{1}{M+1} [s(i) \times g(i) + \sum_{i=0}^{M-1} s(i) \cdot g(i)] \quad (13)$$

where \times is the operation of correlation. The second term in (13) reflects the second term in relation (1).

Without noises the following relation holds [7]

$$\sum_{i=0}^{M-1} f(i) \times g(i) \times g(i) = (M+1) \sum_{i=0}^{M-1} f(i) \quad (14)$$

In the presence of noises relation (14) is valid only approximately in the limits of (12).

If the statistical error of the measurements of the useful events $f(i)$ is negligible, then the total error is determined by the noise from the photomultiplier, and finally

$$f_{red}(i) = \frac{1}{M+1} [s(i) \times g(i) + \sum_{i=0}^{M-1} s(i) \times g(i)] = \frac{1}{M+1} [f(i) \times g(i) + \sum_{i=0}^{M-1} f(i) \times g(i)] + \quad (15)$$

$$+ \Delta v(i) = f(i) + \Delta v(i).$$

Let us compare this result with the traditional system of a homogeneous scintillator. For the latter system

$$f_{trad}(i) = f(i) + \Delta n(i) \quad (16)$$

The gain in effective signal-to-noise ratio is given by [4]

$$\frac{\sigma_{\Delta n}}{\sigma_{\Delta v}} = \frac{M+1}{\sqrt{M}} \quad (17)$$

The corridor of noise errors for the bipolar multiplex transform system is $(M+1)/\sqrt{M}$ times narrower, than for the traditional system.

Results of the Simulation

The simulation accomplished for the case of $M = 19$ clearly demonstrates the effect of suppression of the noise-error. The initial distribution $f(i)$ from $i_{in} = 0$ till $i_{fin} = 18$ is shown in Fig. 5. The bipolar kernel $g(i)$ for $M = 19$ is given in Fig. 6. The integral transform $r(i)$ shown in Fig. 7 is the subject of the measurements. The bipolar multiplex system without noise does sample the function

$$r(i) = f(i) \times g(i) = \sum_{k=0}^{18} f(k) g(i-k) \quad (18)$$

This intermediate data undergo the decoding. The results of the correlation

$$r(i) \times g(i) \quad (1)$$

are shown in Fig. 8. This function differs from the initial function $f(i)$ in two points, namely, in the ratio of scaling and in the different biases.

The noise components $n_1(i)$ from the lower photomultiplier and $n_2(i)$ from the upper photomultiplier are shown in Fig. 9. The bipolar pulse counter detects the fluctuating component of the noise $\Delta n(i)$, shown in Fig. 10. The results of decoding of the fluctuating noise component are given in Fig. 11. The ratio of standard noise errors found for this simulation is equal to $\sigma_{red}/\sigma_{multip} = 4.65$, instead of the theoretical mean value

$$(\sigma_{red}/\sigma_{multip})_{theor.} = \frac{M+1}{\sqrt{M}} = 4.59.$$

References

1. M.H. Fizeau. Ann. Chimie et de Physique, **66**, (1862), 429.
2. M.J.E. Goley. J.O.S.A., **39**, (1949), 437.
3. H.Harmuth. Transmission of Information by Orthogonal Functions. New York/Berlin, 1969, Springer-Verlag.
4. E.D.Nelson, M.L.Fredman, J.O.S.A. **60**, (1970), 1664.
5. A.I. Mogilner, O.A. Salnikov, L.A.Timokhin. PTE, 1966, No 2, 22.
6. W.Glaser und F.Gompf. Nucleonik, **12**, (1969), 153.
7. L.M.Soroko. JINR Commun., Dubna, P13-5696, 1971.
8. L.D.Baumert. Codes with Special Correlation, Ch. 4 in Golomb S.W., Digital Commun., Prentice-Hall, Englewood Cliffs, N.J. 1964.

9. J.A.Decker. Experimental Hadamard-Transform Spectrometry, in 1970 Proceedings Symposium and Workshop on Applications of Walsh Functions, Ed. C.A. Bass, 1970, p. 101.
10. L.R. Welsh. Walsh Functions and Hadamard Matrices, *ibid.* p. 163.
11. J.A.Decker. Applied Optics, 10, (1971), 510.

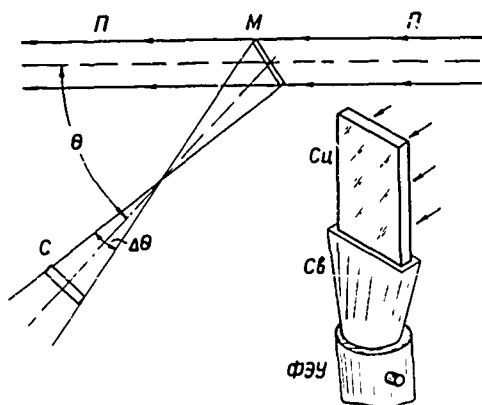


Fig. 1. The traditional arrangement of the particle beam Π , the target M and the counter C of the homogeneous scintillator $C\alpha$ lightguide $C\delta$ and photomultiplier $\Phi Э У$. The particles escaping the target M at the angle θ are registering within the angular resolution $\Delta\theta$.

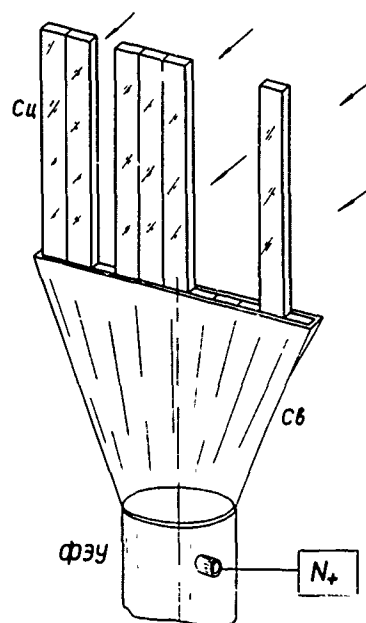


Fig. 2. The unipolar Hadamard transform multiplex counter of several element scintillators (+1) and gaps (0).

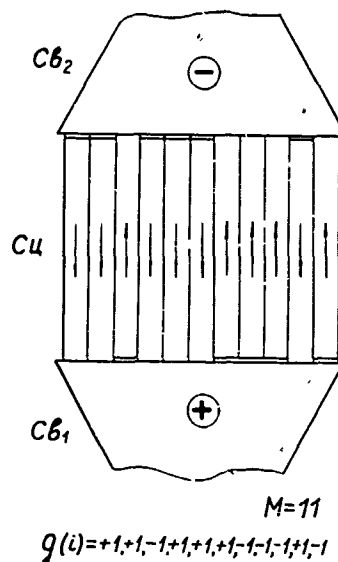


Fig. 3. The bipolar multiplex counter of the two mutually optically isolated combs of element-scintillators. The alternation of the directions (', .) of entering the two lightguides $C\delta_1$ and $C\delta_2$ is described by the bipolar function $g(i)$. The multiplex scintillator shown in Fig. 3 is set up of 11 element-scintillators.

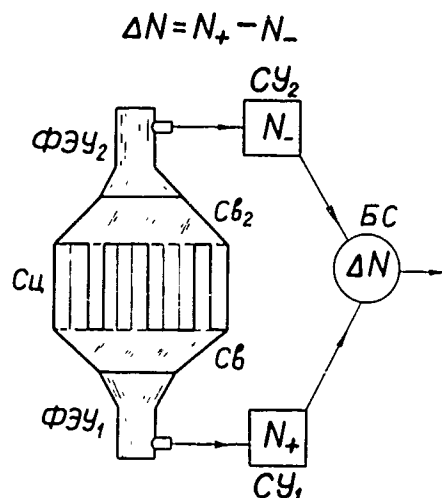


Fig. 4. The electrical scheme includes two photomultipliers $\Phi\Xi Y_1$ and $\Phi\Xi Y_2$, two counting devices CY_1 and CY_2 , and the electrical pulse bipolar counter ΔN .

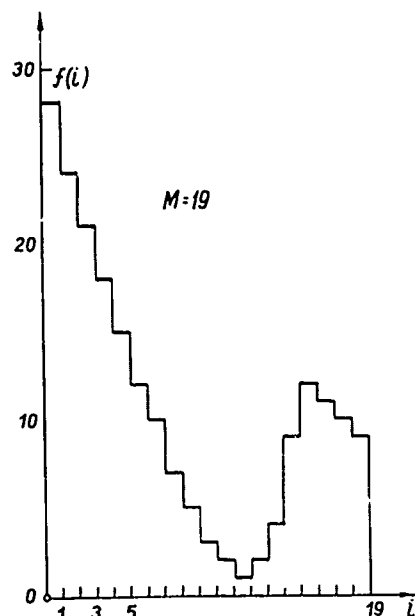


Fig. 5. The initial distribution, $f(i)$ for $M = 19$, used for simulation.

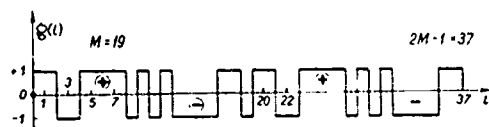


Fig. 6. The bipolar kernel $g(i)$ for $M = 19$.

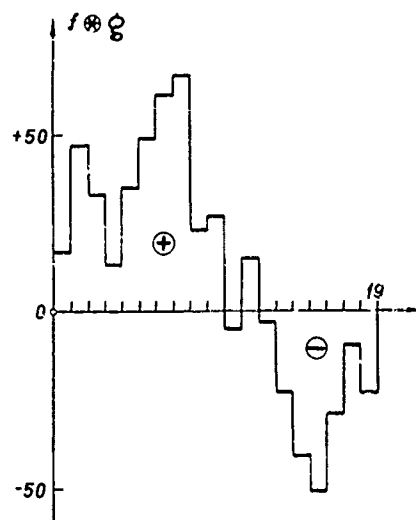


Fig. 7. The integral transform $r(i)$ of the function $f(i)$ shown in Fig. 5.

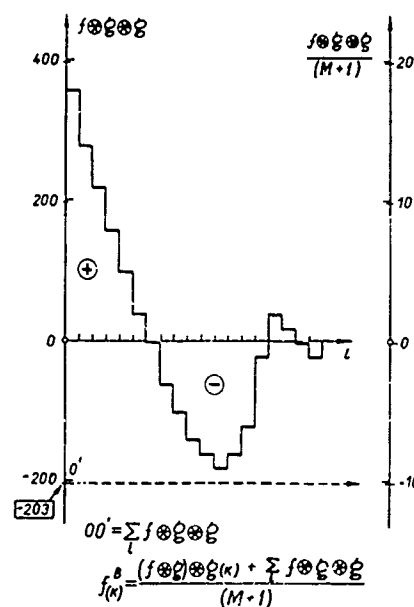


Fig. 8. The results of decoding correlation $r(i) * g(i)$.

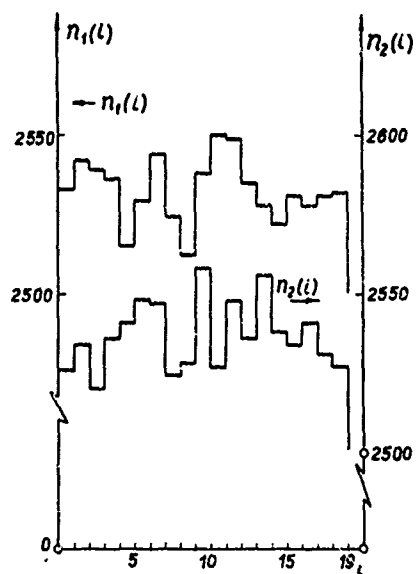


Fig. 9. The noise components $n_1(i)$ and $n_2(i)$ from the lower and from the upper photomultipliers.

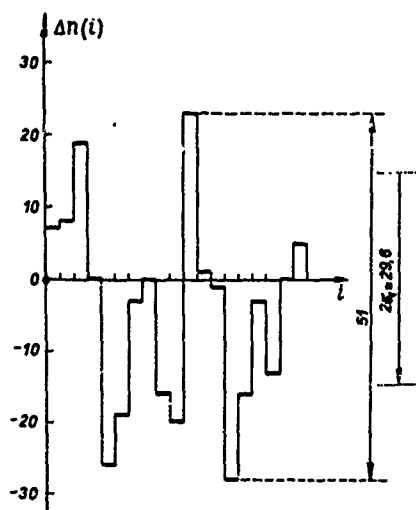


Fig. 10. The fluctuating noise component $\Delta n(i)$.

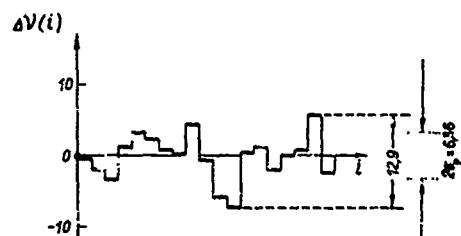


Fig. 11. The decoded fluctuating noise component $\Delta V(i)$.

MULTIPLEX TARGET

by

L.M. Soroko

Joint Institute for Nuclear Research
Dubna, USSR

Abstract

The multiplex Hadamard transform target designed to measure the angular distribution of the particles in the presence of heavy background in the experimental hall is described. The gain in the effective signal-to-background ratio achieved by this multiplex target is the result of spatial multiplexing, that is, of coding the angular distribution to be measured before counting it.

The multiplex target has multi-element heterogeneous structure, which permits the detection of particles at several angles simultaneously by the integral coding algorithm. The transform code is the cyclic trans-orthogonal Hadamard code. The angular distribution of particles to be measured can be found on the second stage by decoding transformation with the same code, as was used on the stage of primary coding.

The multiplex target gives the gain in the signal-to-background ratio if the background does not depend on the parameters of the target and the beam. This gain is equal to $(M+1)/2\sqrt{M}$, where M is the number of comb elements of the multiplex target. The multiplex Hadamard transform target gives the direct efficiency gain in the working time of particle accelerators and also in the cost of the equipment.

Introduction

The traditional target used in the experiments with high energy particles consists of the substance of desirable aggregate state. The target possesses almost homogeneous properties inside the region at which the beam of accelerated particles hit. The sizes and the depth of the target are chosen according to the required energy and/or angular resolution.

The counter or the particle telescope of several counters is commonly hit

not only by the particles which emerge from the target but also by the particles which originate outside the target and are the background ones. At the high level of background particles the error of the measurements is determined almost completely by the background.

Any variations in the geometrical sizes of the counter do not change the relation between the useful events and the background ones. The increase of the target dimensions does deteriorate the spatial or angular resolution of the system.

Multi-Element Target

Meanwhile there is a nontraditional solution of this conflicting situation, namely the use of the multiplex target. By leaving unaltered the spatial resolution of the target we take several elements of target and perform the measurements simultaneously at several angles. Thus, the one and the same background event detected by the counter falls within several angular intervals, but not within a singular angular interval as in the traditional target.

For this aim the target is transformed into heterogeneous multiplex target. The latter has the structure of a comb. The prongs and gaps of the comb alternate one another. The law of this alternation is given by the discrete pseudorandom unipolar sequences $g_+(i)$ or $g_-(i)$. The function $g_+(i)$ can have the values +1 or 0 and the function $g_-(i)$ the values (-1) or 0. The function $g(i)$ which is equal to the sum of $g_+(i)$ and $g_-(i)$ functions forms the row of the truncated Hadamard matrix with the trans-orthogonal properties, that is

$$g(i) \times g(i) = \sum_{k=0}^{M-1} g(i) g(i+k) = \\ = (M+1) \delta(i) - 1(i). \quad (1)$$

where M is the number of elements prongs in comb target.

The angular distribution of the particles emerging from multiplex target undergoes the coding transform. Instead of the function $I(i)$ the counter does sample the function

$$N_1(i) = I(i) \otimes g_+(i) + n_1(i) \quad (2)$$

from the comb target $g_+(i)$ and another function

$$N_2(i) = I(i) \otimes g_-(i) + n_2(i) \quad (3)$$

from the comb target $g_-(i)$. Here $n_1(i)$ and $n_2(i)$ are the number of the background particles hit the counter during the first and the second measurements. The data to be measured in the experiment are equal to the difference

$$\Delta N(i) = N_1(i) - N_2(i) \quad (4)$$

These intermediate data must be retransformed by the decoding correlation with the help of the bipolar $g(i)$ function. The result has the form

$$I_{\text{rest}}(i) = \frac{1}{M+1} [\Delta N(i) \times g(i) + \sum_{j=1}^{M-1} \Delta N(j) \times g(j)] + \Delta v(i) \quad (5)$$

The ratio of the mean square background-error $[\Delta v(i)]_{\text{m.s.}}$ for the multiplex target to the m.s. background-error $[\Delta n(i)]_{\text{m.s.}}$ for the traditional system is equal to $2\sqrt{M}/(M+1)$.

The multiplex target consists of the substance in any aggregate state. There are two sets of the containers or frames for two unipolar multiplex comb-targets. The elements of the combs can move independently relative to the particle beam. Two arrangements of the comb-elements must be built according to the unipolar sequences $g_+(i)$ or $g_-(i)$. The change of the polarity of the bipolar counting device is performed simultaneously with the exchange of one comb-target to another one.

The total number of elements of multiplex targets, including prongs and gaps is equal to $(2M+1)$. These elements are subdivided into three groups, the first group being outside of the beam to the left, the second group of elements hitting by the primary particles of the beam and third group being outside of the beam to the right. In the course of the measurements the spatial structures of the combs undergo the cyclic rearrangements. It is necessary to perform M independent measurements with $g_+(i)$ multiplex target and as many with $g_-(i)$ multiplex target.

The structures of prongs and gaps for two unipolar multiplex targets for the case of $M=7$ are shown in Fig. 1. There are 13 elements which broken down into three groups (Fig. 2). The beam of the particles shown as a ring in Fig. 2 intersects only M elements, prongs or gaps [2]. The configurations of the multiplex targets $g_+(i)$ and $g_-(i)$ are shown for the $i=3$.

The results of the simulation are analogous to the case of $M=19$ [1]. The ratio of the mean square background-error for the multiplex target to the mean square background-error for the traditional system is equal to $2\sqrt{M}/(M+1)$.

References

1. L.M. Soroko. JINR Comm., P13-5696, 1971.
2. L.M. Soroko. JINR Comm., P13-5699, Dubna, 1971.

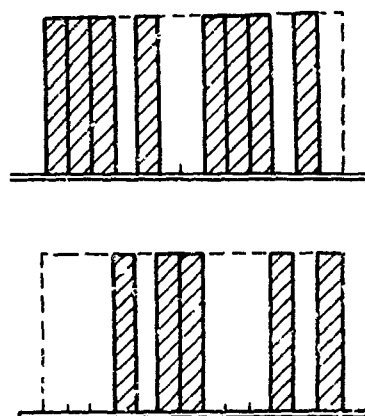


Fig. 1. The spatial structure of multiplex target consisting of two unipolar comb-targets. The measurements are carried out at seven different intervals ($M=7$) simultaneously. The comb-targets consist of 13 elements prongs and gaps, $(2M+1)$. The upper comb-target corresponds to the $g_+(i)$ coding function. The lower one corresponds to the $g_-(i)$ coding function.

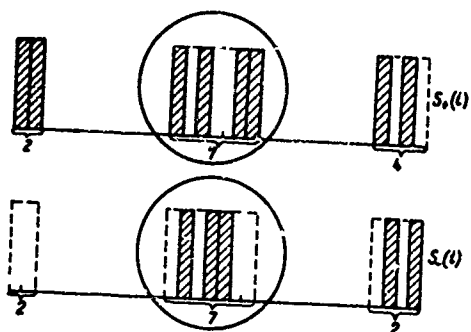


Fig. 2. The arrangement of the elements, prongs and gaps, of the comb-targets relative to the cross section of the beam.

INFORMATION PROPERTIES OF PARTICLE TELESCOPE OF MULTIPLEX SCINTILLATION COUNTERS

by

L.M. Soroko

Joint Institute for Nuclear Research
Dubna, USSR

Abstract

The particle telescope of two multiplex Hadamard transform scintillation counters is described. Each scintillator is set up of several element-scintillators which are surrounded by the light-guides and photomultipliers. Four coincidence circuits are detecting the four groups of events. The coincidence counting rates define the intensity of the angular correlation events induced by the two-particle channels. And the angular resolution of the discrimination of the events to be measured is equal to the angle which subtends the single element-scintillator of the multiplex scintillation counter.

Such a particle telescope allows to solve some new problems, for example, it can be used in the experiments with interesting beams to restrict the volume from which the binary channels particles are emerging and this can be done without the additional defining counters.

Introduction

In the previous communications the Hadamard transform scintillation counter /1/, the multiplex target /2/ and the principle of their operation have been described. The telescope of two multiplex scintillation counters presented earlier in the communication /4/ is reported below.

The traditional telescope designed to detect the penetrating particles is set up of two or more scintillation counter, electrical pulses from which are fed to the coincidence or anti-coincidence circuits. The connections between the telescope counter signals are chosen in such a manner, that the output pulse would come when the light-flashes in either photomultipliers originate simultaneously that is in the time resolution of the coincidence circuit.

Let us consider that traditional telescope of two scintillation counters

(Fig. 1). The pulses from the counters are fed to the coincidence circuit. Such a telescope counts the processes induced in the target by the particles of the beam when one of the secondary particles is emitted at the angle θ , and the other one is emitted at the angle $(180^\circ - \theta)$ to the beam axis. To detect two-particle channels in the wide interval of angles from θ_1 to θ_2 , one does the following. A sufficient small target and a sufficient small telescope angular view are taken, so that the halfwidth of the angular correlation curve has the required value $\Delta\theta$ (Fig.2). Under real conditions the maximum of the curve lays above the pedestal due to the background events from many particle channels. The measurements are made at several angles θ from θ_1 to θ_2 . The required results can be obtained by summation of the events which are above the pedestal. Any attempts to increase the telescope angular view give the broadening of the maximum and a higher pedestal.

Telescope of Multiplex Scintillation Counters

Meanwhile there is a nontraditional way to cover all the correlation events within the wide interval of angles from θ_1 to θ_2 , without the loss of angular correlation resolution. The telescope which processes these properties is of the following design. The homogeneous scintillators in the counters are exchanged by the heterogeneous multiplex scintillators /1/. Each multiplex scintillator is set up of several element-scintillators (Fig. 3). The butt-ends of the multiplex scintillators are surrounded by lightguides and photomultipliers. The light-flashes induced by the charged particles hit the element-scintillators can enter only one of two lightguides and can not enter the adjacent element-scintillator. The direc-

tions of the light-outcomes alternate according to the law of pseudorandom coding ^{/3/}. In the case of seven element-scintillators ($M = 7$), this alternation of the light-outcomes is described by the rule

for the first scintillation counter, and by the "mirror" rule

for the second scintillation counter.

The electrical pulses from four photomultipliers are fed to four coincidence circuits (Fig. 4), thus permitting to count four groups of events:

N_{++} , N_{--} , N_{+-} and N_{-+} . These observed counting rates, as shown in ^{/4/}, completely define the counting rate of the correlation events to be searched. The angular correlation resolution of the events with 180° correlation angle is defined by the angle which subtends the single element-scintillator of the multiplex scintillation counter, if the dimensions of the target being very small. The number of the 180° -correlation events to be searched is equal

$$N_{corr} = (N_{++} + N_{--}) - \frac{M^2 + 1}{M^2 - 1} (N_{+-} + N_{-+}) \quad (1)$$

The background correlation events from many particle channels are supposed to be distributed uniformly.

It is easily seen that this algorithm is valuable not only for the case of the 180° correlation angle, but also for any correlation angle. The correlation angle is changing by the turn of the telescope arm, as in the case of the traditional telescope. It is possible also to shear the coding pictures of the counters relative to each other. The processing algorithm for the latter case has the form different from (1).

New Problems

The telescope of multiplex scintillation counters allows to solve some new problems. Namely, such a telescope can be used in the experiments with intersecting beams to restrict the volume from which the binary channel particles are emitted, and this can be done without the additional defining counters. Such an experiment is shown in Fig. 5. The telescope of two multiplex scintillation counters C_1 and C_2 is turned to count the binary channel particles emerging from the cell with number $k=0$.

The algorithm of the processing is defined by (1). The background events include the particles from binary pro-

cesses which taking place in the cells with the numbers $k \neq 0$, and also the uniformly distributed particles from many-particle channels.

The telescope of multiplex scintillation counters allows to detect the particles with given curvature on the background of the particles with different values of curvature. Another examples are as follows. 1) The restriction of the working volume inside the gaseous target without the help of the collimating entrance slits or defining counters. 2) The detection of the events in the target, immersed in the constant magnetic field. The events to be measured are characterized by the combination of two parameters: the curvature of the trajectory and angular correlation between particles ^{/4/}.

References

1. L.M. Soroko. JINR Comm., Dubna, P13-5696, 1971.
2. L.M. Soroko. JINR Comm., Dubna, P13-5699, 1971.
3. L.M. Soroko. JINR Comm., Dubna, P13-5722, 1971.
4. L.M. Soroko. JINR Comm., Dubna, P13-5896, 1971.

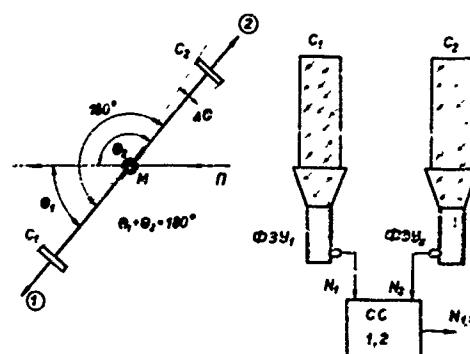


Fig. 1. Traditional telescope of two scintillation counters C_1 and C_2 . The particle correlation angle is equal to 180° .

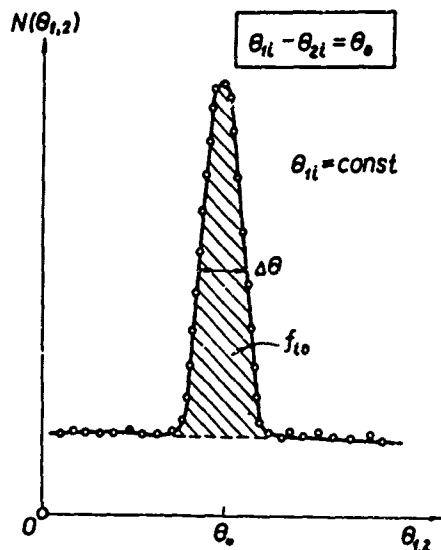


Fig. 2. The angular correlation curve measured by the traditional telescope as a result of the movement of the one counter. The mean correlation angle is equal to θ_0 and the angular resolution is equal to $\Delta\theta$.

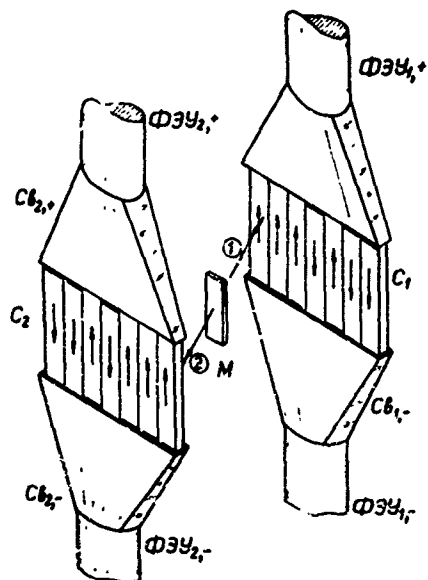


Fig. 3. The telescope of two multiplex scintillation counters and the target M . The telescope is set up of two multiplex scintillators C_1 and C_2 , four light-guides and four photomultipliers. The alternation of the light outcomes is central symmetrical.

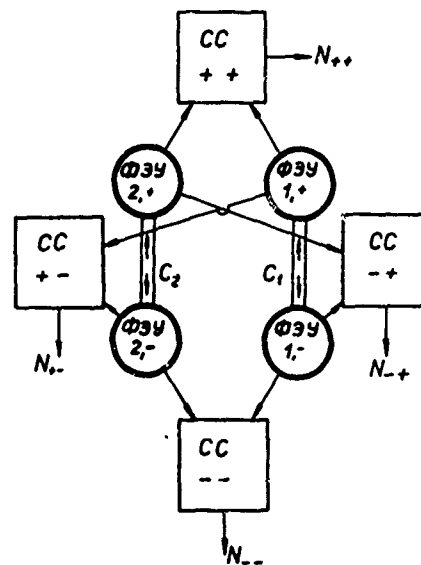


Fig. 4. Four coincidence circuits coupled with four photomultipliers of the telescope.

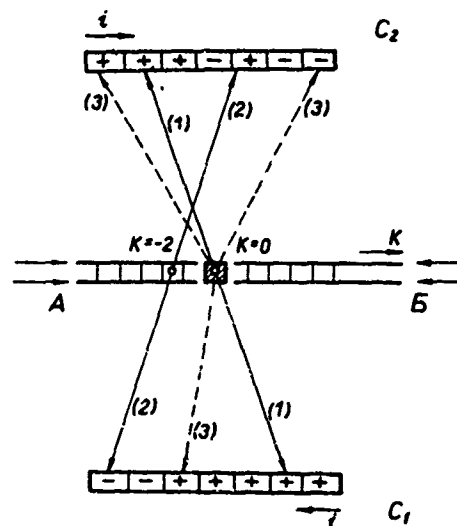


Fig. 5. The experiment with intersecting beams. The cells of the common volume are numerated by the index K .

WALSH SPECTROSCOPY OF RAYLEIGH WAVES CAUSED BY UNDERGROUND DETONATIONS

M. Båth
Seismological Institute
Uppsala
Sweden

and

S. Burman
Council of Scientific and Industrial Research
New Delhi
India

SUMMARY

There are increasing attempts to apply Walsh functions to communication engineering and other fields. This paper describes the results of Walsh and Fourier spectra of the vertical components of Rayleigh waves at a set of stations, viz., Oulu, Uppsala, Copenhagen and Stuttgart, due to the underground nuclear explosion at Novaya Szezlya on Oct. 27, 1966. It was found that both the Fourier and the Walsh spectra at all these stations have interference peaks and troughs as well as sharp cutoffs at both the low and high frequency ends. Most of the energies in these stations are concentrated in the 0.08 - 0.2 cps (Fourier) and 0.08 - 0.2 cps (Walsh) range. An interesting feature in the Walsh case is that the spectral heights can be put into two groups, M, the primary one at the lower frequency range and N, the secondary one at the higher end. Most of the energy is in the M group in case of Walsh spectrum.

In both cases, for the farther stations the spectra shifts toward the lower frequency end. The physical reason may be due to the propagation and/or to the gradual lengthening of the wave period. Because of the complicity in the mathematical formulation of arithmetical convolution in case of Walsh transform, it is difficult to get true ground motion spectra from the recordings. Since, however, the Walsh spectrum obtained does characterise the signal, it may have some use as a simply performed

operation to get a general indication of the spectral nature of the signal.

INTRODUCTION

There is an upsurge of interest presently in engineering circles on the use of functions, such as the Walsh functions, with orthogonal series other than the traditional sine-cosine series. The present paper compares the Walsh and the trigonometric Fourier power spectra of a set of Rayleigh waves (using vertical components) from the underground nuclear explosion on Oct. 27, 1966 by the Russians at Novaya Szezlya recorded at several stations, viz., Oulu, Uppsala, Copenhagen and Stuttgart lying on or very near the great circle path through the explosion site and the stations.

The Walsh function set, being a binary one, is directly suitable with digital computers and digital circuitry. In contrast, when the Fourier transformation by digital computer as the decoding process is considered, it is known that it falls short of being ideal because the trigonometric functions sine and cosine involved in it are not naturally fitted to the essentially binary operations of digital computers. Another advantage of using Walsh function set is that it is a somewhat natural describer of pulse or pulse-like signals (Brown, 1970); that makes it suitable for studying pulse-like seismic signals such as the P waves.

Karmuth (1968, Equation 2) has given the Walsh-Fourier series expansion of a function $F(x)$ defined in the interval $-Y/2 \leq x < Y/2$, as

$$F(x) = a(0)\text{wal}(0,x) + \sum_{i=1}^{\infty} [a_c(i)\text{cal}(i,x) + a_s(i)\text{sal}(i,x)]$$

where,

$$\begin{aligned} a(0) &= \int_{-Y/2}^{Y/2} F(x)\text{wal}(0,x)dx \\ &= \int_{-Y/2}^{Y/2} F(x)dx \\ a_c(i) &= \int_{-Y/2}^{Y/2} F(x)\text{cal}(i,x)dx \\ a_s(i) &= \int_{-Y/2}^{Y/2} F(x)\text{sal}(i,x)dx \end{aligned} \quad \dots (1)$$

Squaring the Walsh-Fourier series expansion (eqn. 1) and integrating over the interval of orthogonality, $-Y/2 \leq x < Y/2$, one gets,

$$\begin{aligned} \int_{-Y/2}^{Y/2} F^2(x)dx &= Y T \int_{-Y/2}^{Y/2} F^2(t/T)dt \\ &= a^2(0) + \sum_{i=1}^{\infty} [a_c^2(i) + a_s^2(i)] \end{aligned} \quad \dots (2)$$

where, $x = t/T$, T is the time unit used to normalise the time variable t in $\text{cal}(i,x)$ and $\text{sal}(i,x)$. The left side of eqn. 2 is interpreted as the average power of a signal $F(x)$.

$a^2(0)$ and $a_c^2(i) + a_s^2(i)$ are then interpreted as the power spectrum of the discrete variable i .

For the purpose of computation the Finite Walsh Transform theory and the Fast Walsh Transform algorithm developed by Kennett (1970a, 1970b) were used. There exists a convenient representation of the discrete Walsh functions $\text{wal}(k,j)$ in terms of the binary representation of the indices; the N -length discrete Walsh functions may be defined for $N = 2^p$ by a continued product representation (Kennett, 1970a, 1970b).

For an N -length real sequence $X(j)$ where $j = 0, 1, 2, \dots, N-1$, the finite Walsh transform is defined as

$$\bar{X}(k) = (Y/N) \sum_{j=0}^{N-1} X(j)\text{wal}(k,j) \quad \dots (3)$$

where, $N = 2^p$; $p = 1, 2, \dots$; $k = 0, 1, 2, \dots, N-1$ and $\text{wal}(k,j)$ is the N -length discrete Walsh function of order k . Similarly, $X(j)$ may be expressed as the inverse finite Walsh transform of the sequence $\bar{X}(k)$,

$$\begin{aligned} X(j) &= \sum_{k=0}^{N-1} \bar{X}(k)\text{wal}(k,j) \\ &= \sum_{k=0}^{N-1} \bar{X}(k)\text{wal}(j,k) \end{aligned} \quad \dots (4)$$

Kennett's paper (1970a) contains the Fortran program for computing the Fast Walsh transform for $N = 128$ i.e., for $p = 7$. This program calculates the expression under the summation sign in eqn. 3 namely,

$$\sum_{j=0}^{N-1} X(j)\text{wal}(k,j)$$

and to get the Walsh transform the values are to be divided by N . The factor Y/N is not of significance as in the case of the Fourier transform as one may also define the forward Walsh transform as

$$\bar{X}(k) = \sum_{j=0}^{N-1} X(j)\text{wal}(k,j)$$

and the inverse Walsh transform as

$$X(j) = (Y/N) \sum_{k=0}^{N-1} \bar{X}(k)\text{wal}(k,j)$$

For application to our seismograms the value of p was taken to be 9. The actual number of data points were less than $N = 512$ (being 2 raised to the power 9) points, so that the requisite number of zeros were added at the end of data to make the total number of points equal 512. Kennett's program was broken up into constituent blocks and by the addition of similar blocks was extended for $N = 2^9$ data points. The extended program was tested for $F(x) = 10/x$ and the result was found to be identical to that given by him (1970a). A second test was successfully carried out by calculating the Walsh transform of a sine wave with two cycles.

WALSH & FOURIER SPECTRA

For our work, part of the Rayleigh

wave train from the vertical component of the seismometers due to the underground nuclear explosion at Novaya Ssemlya on October 27, 1966 were taken. The stations selected were Oulu, Uppsala, Copenhagen and Stuttgart -- all lying nearly on a great circle passing through the explosion site and the stations. The seismogram trace amplitudes were taken and the instrument characteristics and the processing of data prior to the computer analysis are described in a separate paper (Bath and Burman, unpublished). It may be mentioned that the rate of digitisation was taken to be one sample per second and a sampled-data Butterworth filter (3 sec - 133 sec) was used to cut-off frequencies both from the higher and lower sides.

Figure 1 shows a filtered Rayleigh wave train recorded by the vertical component Press-Ewing seismometer at Uppsala. Figures 2, 3, 4, and 5 depict the Fourier power spectra of Rayleigh waves recorded at Oulu, Uppsala, Copenhagen and Stuttgart respectively. Figures 6, 7, 8 and 9 indicate the Walsh power spectra at those stations taken in the same order as the Fourier spectra.

All the Fourier power spectra are marked by sharp cutoff at both ends of the frequency scale, and also these are characterised by sharp amplitude upwings and drops, probably due to multipath interference and other effects. The spectra at Oulu and Uppsala lying on a homogeneous geologic formation can more easily be correlated. An interesting observation is the shift in the spectrum at Uppsala towards the low-frequency end such as the shift in the peak identified in both by the letter A. This shift may be due to the propagation and/or to the gradual lengthening of wave period. It is of interest to note that the energy content at higher frequencies (from about 0.2 to 0.3 cps) at Uppsala is more than that at Oulu. Although the reason is not presently clear, it is not due to any background noise. Microseismic spectra just before the arrival of the explosion energy at Uppsala were computed and it is insignificant; this is also borne out by visual examination of the record.

The interpretation of the trace

amplitude spectra at Copenhagen is complicated by the fact that a different instrument (Galitsin) with different response characteristics was used. Besides, the spectral response at Copenhagen and Stuttgart were influenced by other factors, such as, (a) deviations from the great circle path, (b) departure from lateral homogeneity, and (c) station factor. The peak A, at the low-frequency cutoff region, has shifted following the previous observation, at Stuttgart in comparison to that at Uppsala.

The Walsh -Harmuth power spectra at all these four stations have interference peaks and troughs as well as sharp cutoffs at both low and high frequency ends as in the case of the Fourier ones. Most of the energy is concentrated in the range 0.08-0.2 cps, and in case of the Fourier power spectra it is approximately the range 0.08-0.2 cps. Of great interest in these Walsh spectra is the observation that the spectral heights can be put in two groups, M, the primary at the lower frequency range, and N, the secondary at the higher end. Most of the energy is in the M group. For comparison it is interesting to study the Fourier and Walsh spectra due to a damped monochromatic sinusoidal wave (Campanella & Robinson, 1970) which is shown in Figure 10. The Fourier amplitude spectrum shows only one maxima while the Walsh one indicates two maximas corresponding to the two groups M and N; the M group containing more energy compared to the N one. Similarly, the M groups in the Walsh spectra of the seismograms represent greater energy than the N ones do.

Kennett (personal communication) has given an explanation of the effects leading to the splitting of the Walsh power spectrum into M, N peaks. From the logical analog of the Wiener - Khinchine theorem (Kennett, 1970a, 1970b) the Walsh power spectrum is the finite Walsh transform of the logical autocorrelation given by

$$L_N(j) = (1/N) \sum_{m=0}^{N-1} X(j \oplus m) \quad \dots (5)$$

where \oplus indicates addition module 2.

Even for large j because of the nature of the addition module 2 some $L_n(j)$, according to Kennett, will only be a measure of short range correlation and so $L_n(j)$ will be a mixture of samples of long and short range correlation in the record- which will require higher order Walsh components to describe them - the M series. For smaller j Kennett thinks this problem will not be so acute and the M series will come from the smoothed out logical autocorrelation sequence.

Comparison of Walsh spectra at different stations also as in the previous case show the shift towards low frequency end of the spectra for the farther stations. The spectral peaks J, K, and L in the region M at Uppsala show shifts towards the low frequency end compared to the corresponding peaks at Oulu. Correlation of the individual peaks become difficult in case of Copenhagen both because of (a) different instrument used, (b) station factor, the nature of the basement rock on which the instrument rests, and (c) departure from lateral homogeneity and for Stuttgart because of the last two factors (b) and (c). In conformity to the region M the peak power values at P, R, and S in the region N at Uppsala indicate, compared to those at Oulu, a shift towards the zero frequency end. Comparison with the other two stations becomes difficult because of the uncertainty in identifying the corresponding peaks.

CONCLUDING REMARKS

In case of the Fourier trace amplitude spectra it is a straightforward procedure to get the true ground motion spectrum by correcting for the response characteristics of the instrument. This follows from the well-known relation that the Fourier transform of the convolution of two signals is equal to the product of the Fourier transforms of the two signals. Such a simple relationship between convolution and the product of Walsh transforms of two sequences does not exist (see equation 8 below, Kennett, 1970b) making it difficult to remove the effect of the instrument characteristics.

Given a sequence $X(j)$ and its

Walsh transform $\bar{X}(k)$ and another sequence $Y(j)$ and its Walsh transform $\bar{Y}(k)$, such that,

$$X(j) \leftrightarrow \bar{X}(k) \text{ and } Y(j) \leftrightarrow \bar{Y}(k) \quad \dots (6)$$

the convolution of these two sequences $X(j)$ and $Y(j)$ is defined by the relation,

$$Z_n(s) = (1/N) \sum_{r=0}^{N-1} X(r)Y(s-r)$$

where,

$$Y(m) = 0,$$

if $m < 0$, and $s = 0, 1, \dots, N-1$.
 $\dots (7)$

Thus,

$$\begin{aligned} Z_n(s) &= (1/N) \sum_{r=0}^{N-1} X(r) \cdot \left[\sum_{k=0}^{N-1} \bar{Y}(k) \text{wal}(k, s-r) \right] \\ &= (1/N) \sum_{k=0}^{N-1} \bar{Y}(k) \cdot \left[\sum_{r=0}^{N-1} X(r) \text{wal}(k, s-r) \right] \\ &= (1/N) \sum_{k=0}^{N-1} \bar{Y}(k) \sum_{r=0}^{N-1} \sum_{k'=s}^{N-1} \bar{X}(k') \cdot \text{wal}(k', r) \text{wal}(k, s-r) \\ &\dots (8) \end{aligned}$$

As seen from Equation (8) the theoretical complication arises from the lack of a known relation between the two functions

$$\text{wal}(k', r) \text{ and } \text{wal}(k, s-r)$$

The same consideration prevents any simple expression for autocorrelation or crosscorrelation in terms of the Walsh transform coefficients.

On the other hand, given $X(j) \leftrightarrow \bar{X}(k)$, $Y(j) \leftrightarrow \bar{Y}(k)$, the logical or dyadic convolution (Gibbs, unpublished report, 1968, Kennett, 1970b) is defined as,

$$Z_1(s) = (1/N) \sum_{r=0}^{N-1} X(r)Y(s \oplus r) \quad \dots (9)$$

Symbolically, it is denoted by,

$$Z_1 = X \odot Y \quad \dots (10)$$

Now,

$$\begin{aligned} Z_1(s) &= (YN) \sum_{r=0}^{N-1} X(r)Y(s \odot r) \\ &= (YN) \sum_{r=0}^{N-1} X(r) \left[\sum_{k=0}^{N-1} \bar{Y}(k) \text{wal}(k, s \odot r) \right] \\ &= (YN) \sum_{k=0}^{N-1} \bar{Y}(k) \left[\sum_{r=0}^{N-1} X(r) \text{wal}(k, r) \text{wal}(k, s) \right] \\ &= \sum_{k=0}^{N-1} \bar{X}(k) \bar{Y}(k) \text{wal}(k, s) \quad \dots (11) \end{aligned}$$

Thus, the logical convolution is the finite Walsh transform of the product of the Walsh transforms of two sequences; symbolically,

$$X \odot Y \longleftrightarrow \bar{X} \cdot \bar{Y} \quad \dots (12)$$

A close similarity is thus seen in the mathematical structure between the arithmetical convolution in case of Fourier and logical convolution in case of Walsh transform theory. However, because of the fact that the trace amplitude in a seismogram is the result of arithmetical convolution between the ground motion and the instrumental characteristics, one is not able to use the simple relation given by the Equation (12) and use of the arithmetical convolution expressed in Equation (8) is complicated, as seen before, by the lack of a known relation between the two functions,

$$\text{wal}(k^1, r) \text{ and } \text{wal}(k, s-r).$$

It is true that by the procedure adopted here one has not obtained a true Walsh power spectrum of the ground motion. Since, however, the spectrum obtained does characterise the signal it may have some use as a simply performed operation to get a general indication of the spectral nature of the signal.

Based on an analogy of detection

of radar signals, Harmuth (personal communication) has pointed out another interesting possible application of Walsh functions to seismology and seismic prospecting. A standard way to detect radar return signals is by means of the autocorrelation function. For long signals the autocorrelation function has a peak value when the reflected signal and a locally produced signal of the same shape have no time shift between each other. However, the sidelobes occur for certain time shifts. If only one radar target is present this causes no problem. If the signal is reflected by several targets it is difficult to decide whether a certain peak in the autocorrelation function is the main peak from a second, third, etc. target or a sidelobe of the first target. By means of Walsh functions one can completely eliminate, according to Harmuth, the sidelobes and thus obtain a definite resolution of several targets that are close together. He envisions that the same technique should be successful in the seismic problem where a signal is multiplied reflected (thereby giving several return signals) through several closely spaced layers in the earth.

ACKNOWLEDGEMENT

We are thankful to Dr. H.F. Harmuth of the University of Maryland for his helpful suggestions and keen interest taken in this work, to Mr. B.L.N. Kennett of the University of Cambridge, England for kindly sending his report of the Walsh Transform which includes the details of the Fast Walsh Transform computational steps, and to Mr. Rolf Svanström, Elema-Schönander AB, Solna, Stockholm for programming help. One of the authors (S.B.) thanks the Swedish I.D.A. for the grant of a fellowship and the Council of Scientific and Industrial Research, New Delhi for the grant of leave of absence.

REFERENCES

1. Brown, C.G., 1970, Signal processing techniques using Walsh Functions, Symposium of the Application of Walsh Functions, Washington, D.C., 138-46.
2. Båth, M and Burman, S.D., (unpubli-

- shed), On the Rayleigh wave attenuation from Fourier amplitude spectra (to be sent for publication to Pure and Applied Geophysics, Milan, Italy).
3. Campanella, S.J. and Robinson, G.S., 1970, Digital sequency decomposition of voice signals, Symposium on the Application of Walsh Functions, Washington, D.C., 230-37.
 4. Gibbs, J.S., 1968 (unpublished), Walsh spectrometry, a form of spectral analysis well suited to binary digital computation: National Physical Laboratory, Teddington, Middlesex, England Report.
 5. Harmuth, H.F., 1968, A generalised concept of frequency and some of its applications: IEEE Transactions on Information theory, IT-14, No. 3, 375-82.
 6. -----, 1971 (personal communication).
 7. Kennett, B.L.N., 1970a (unpublished), The Walsh Transform: British Petroleum Research Center, Middlesex, England Report.
 8. -----, 1970b, A note on the Finite Walsh transform: IEEE Transactions on Information theory, v. IT-16, 489-91.
 9. -----, 1971 (personal communication).

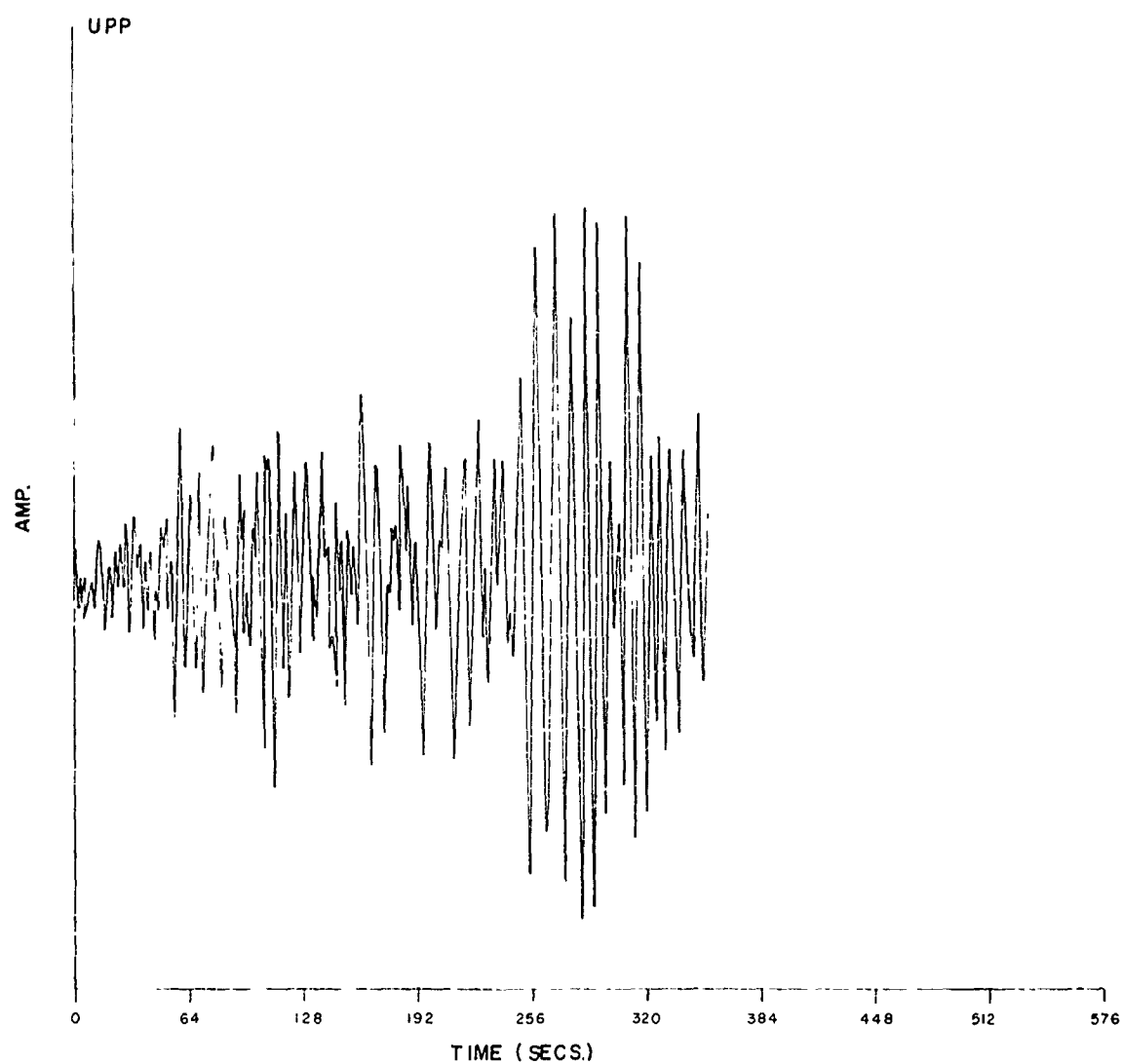


Fig. 1 Filtered Rayleigh wave train.

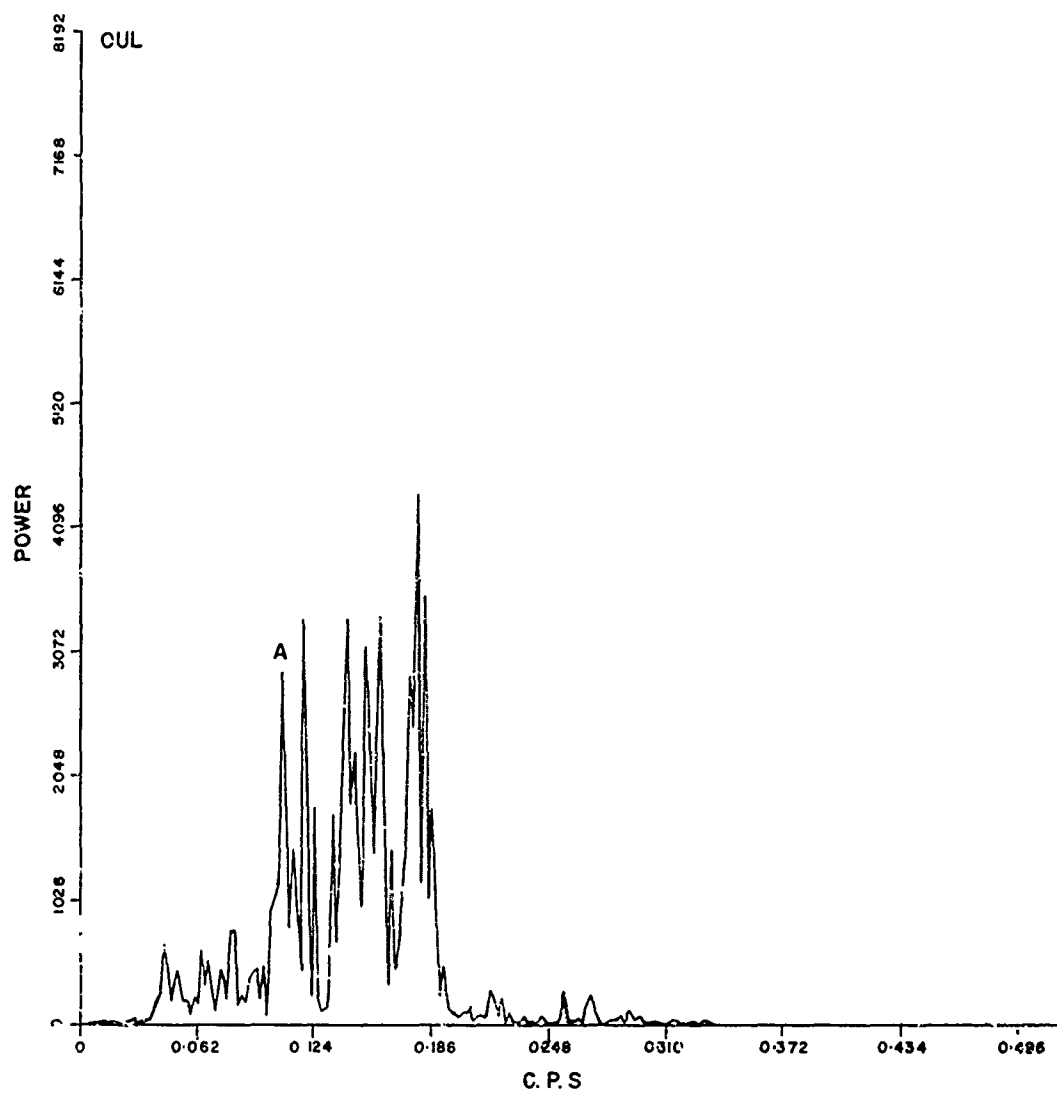


Fig. 2 Frequency power spectrum recorded at Oulu.

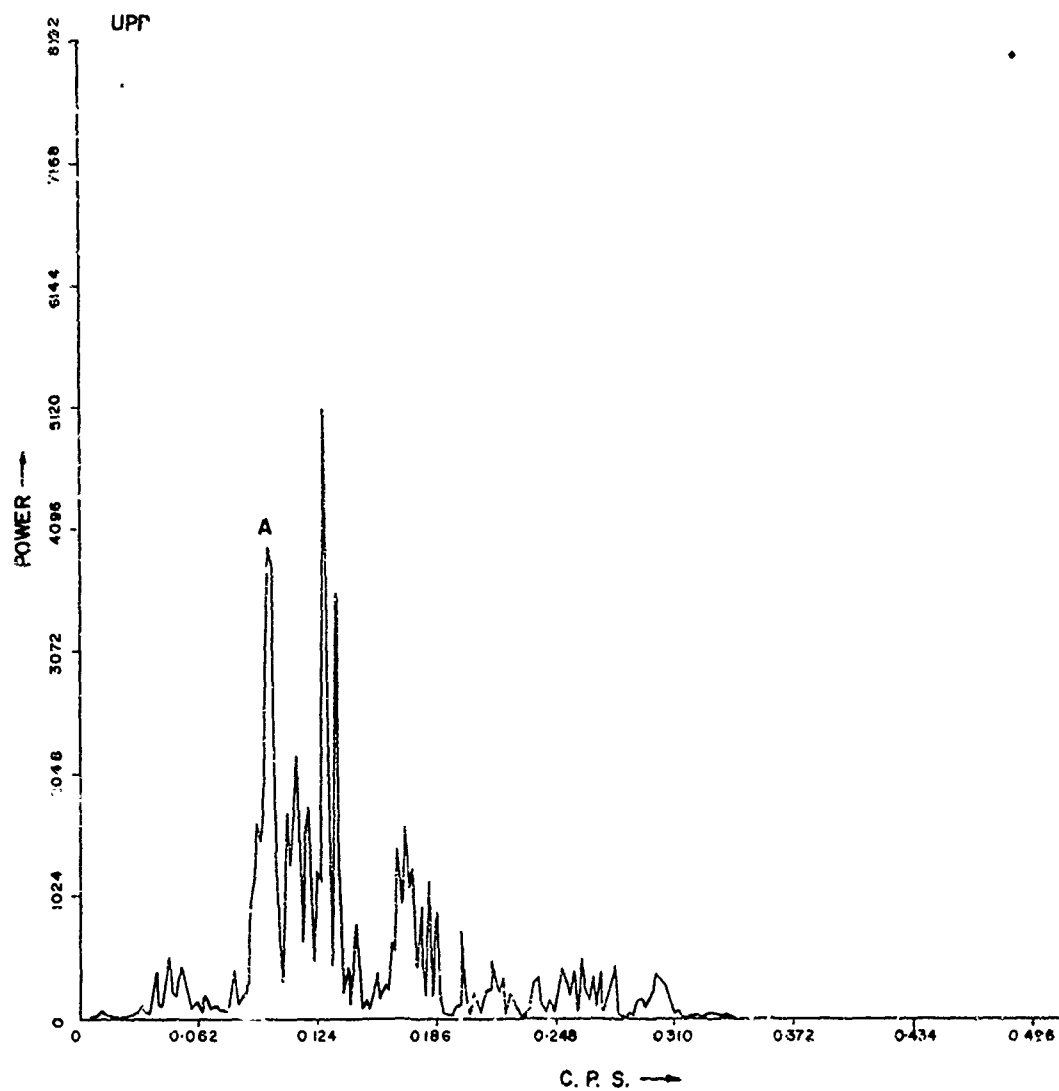


Fig. 3 Frequency power spectrum recorded at Uppsala.

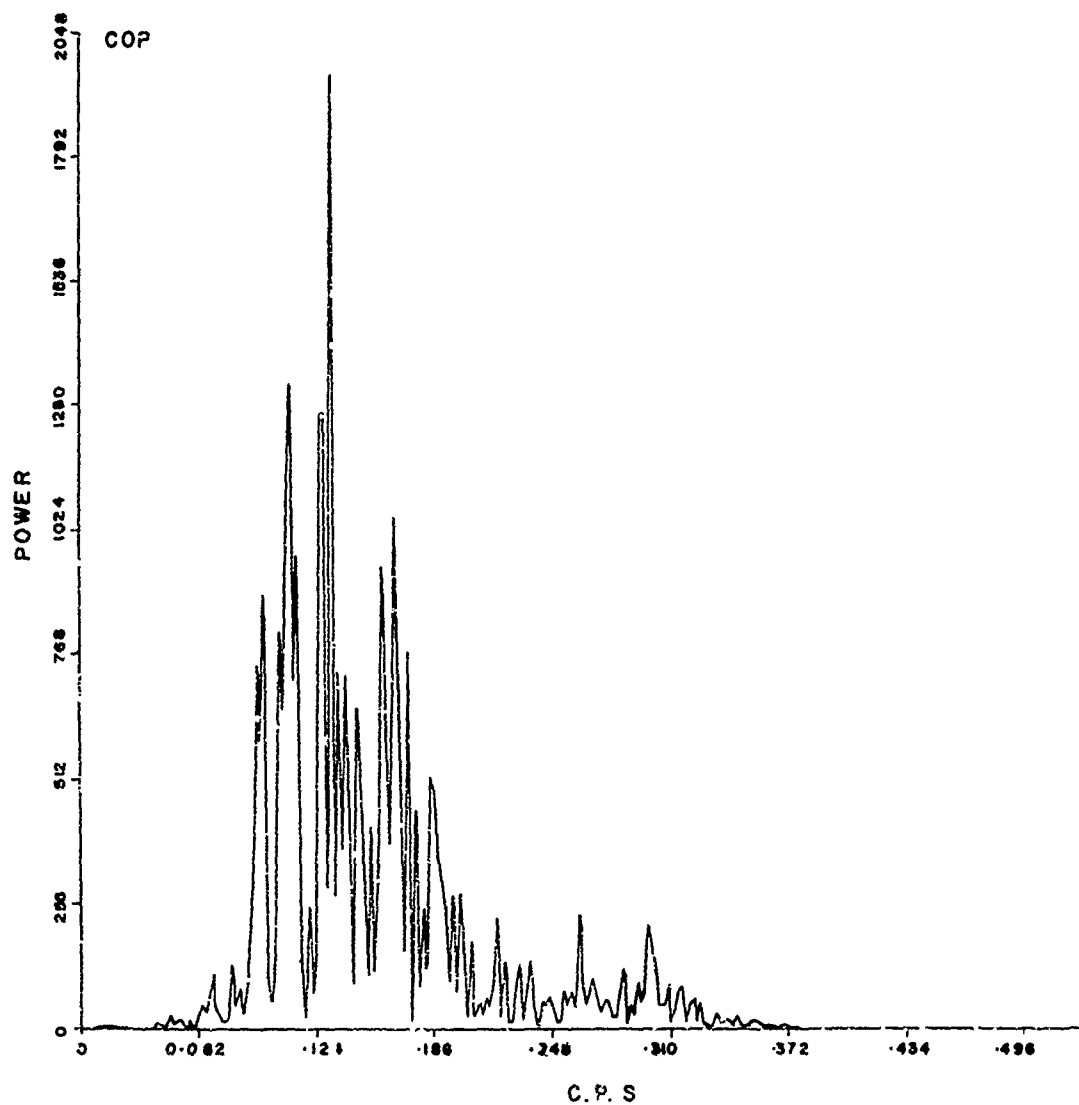


Fig. 4 Frequency power spectrum recorded at Copenhagen.

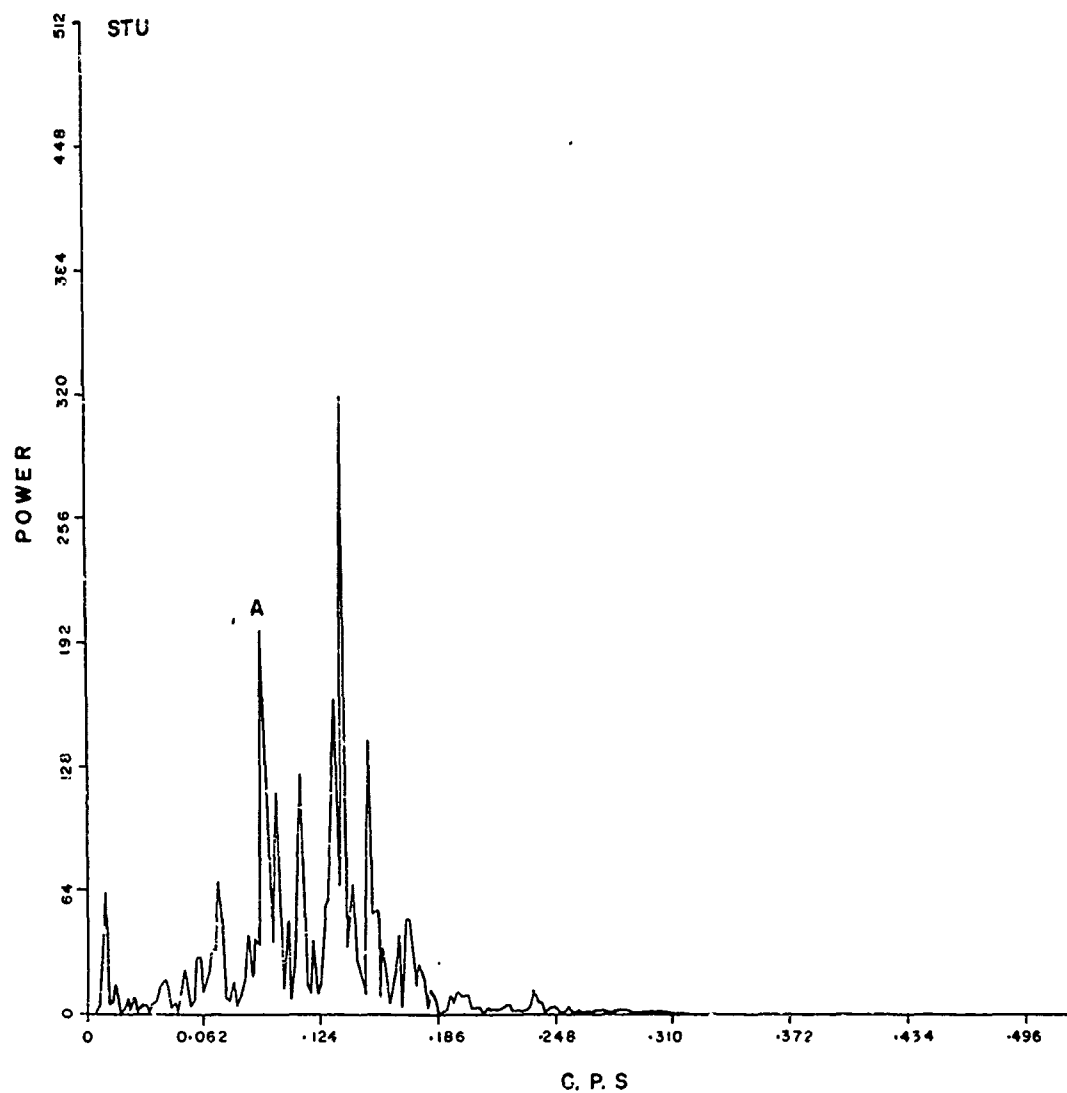


Fig. 5 Frequency power spectrum recorded at Stuttgart.

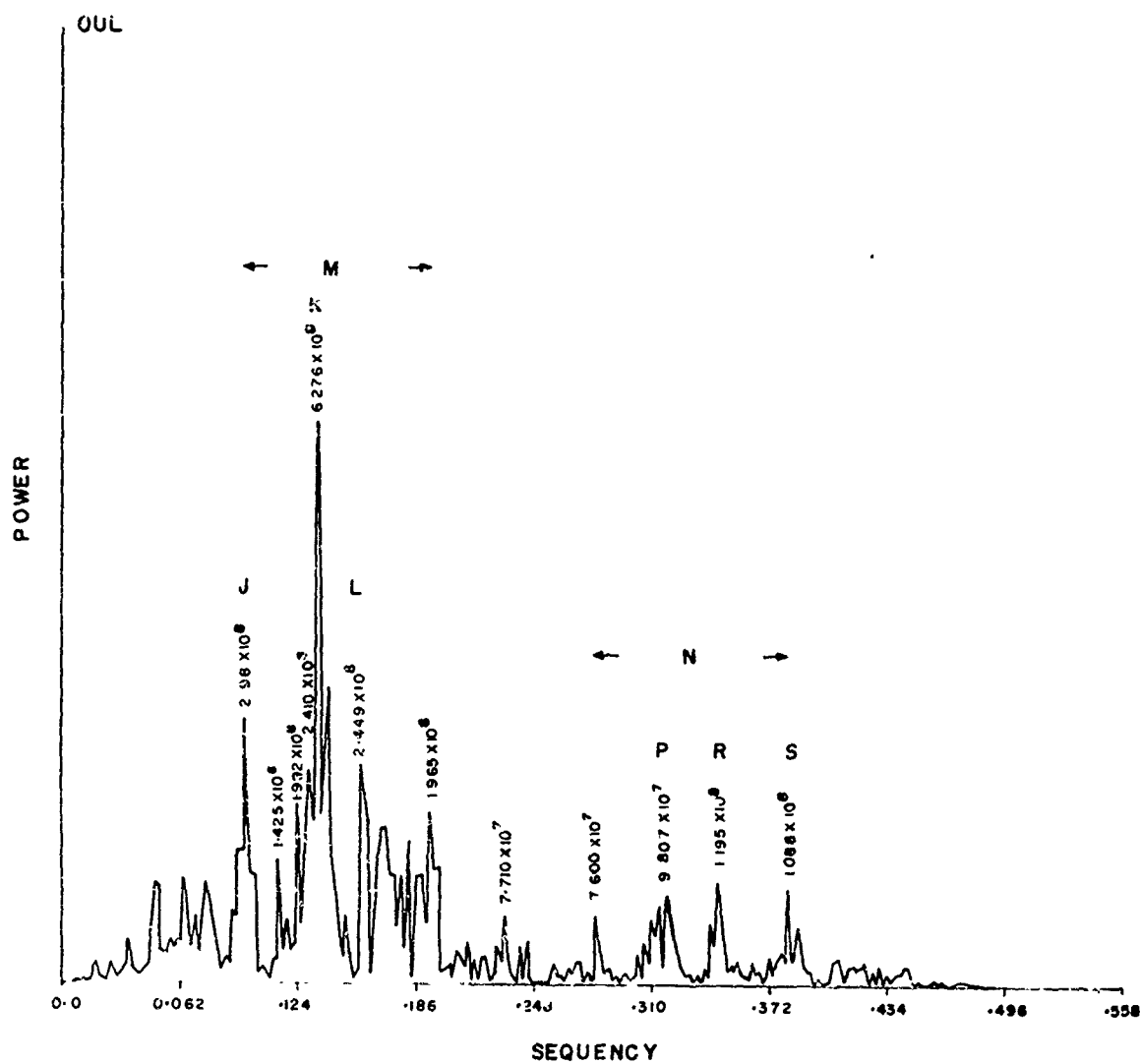


Fig. 6 Sequency power spectrum recorded at Oulu.

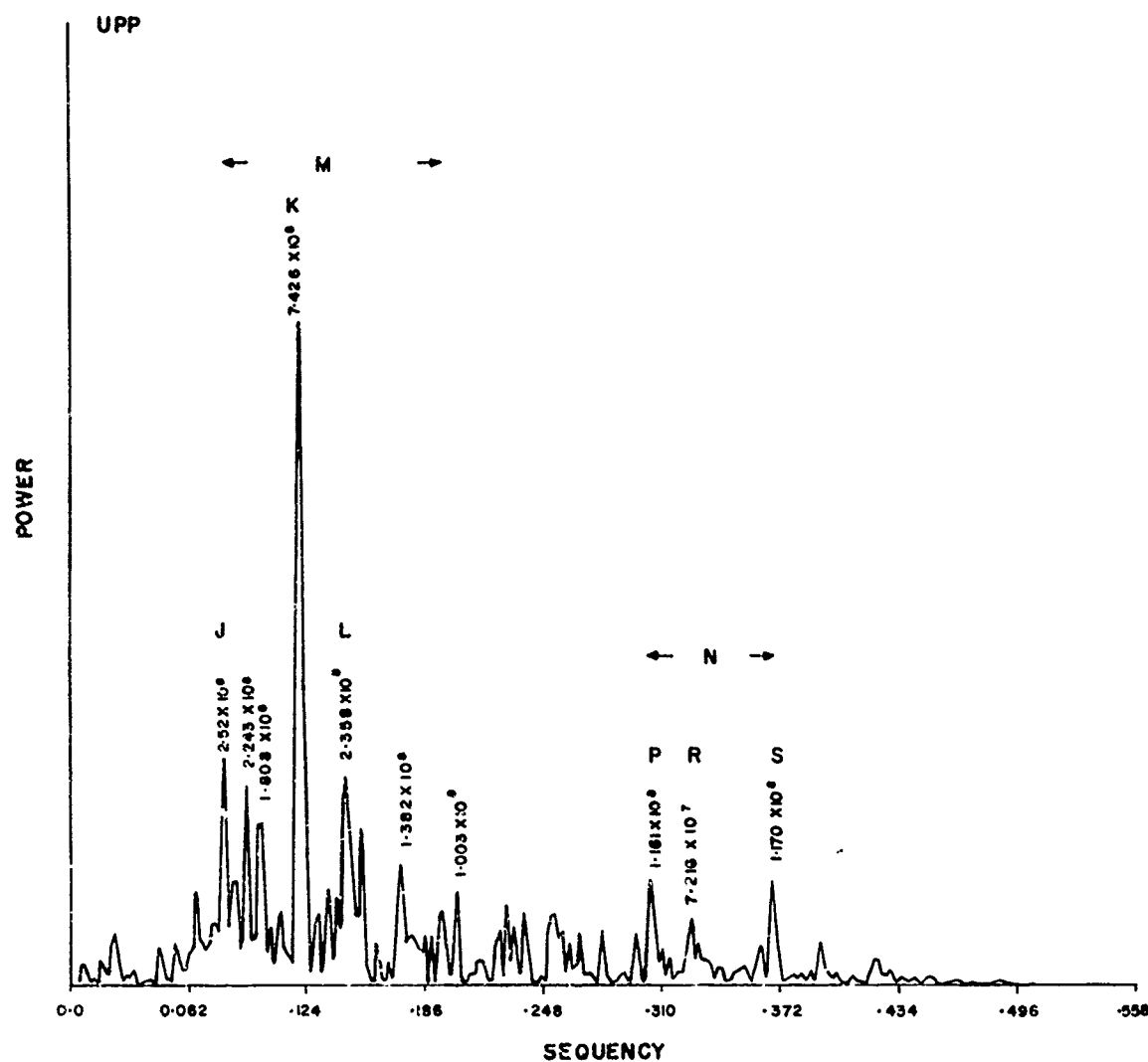


Fig. 7 Sequency power spectrum recorded at Uppsala.

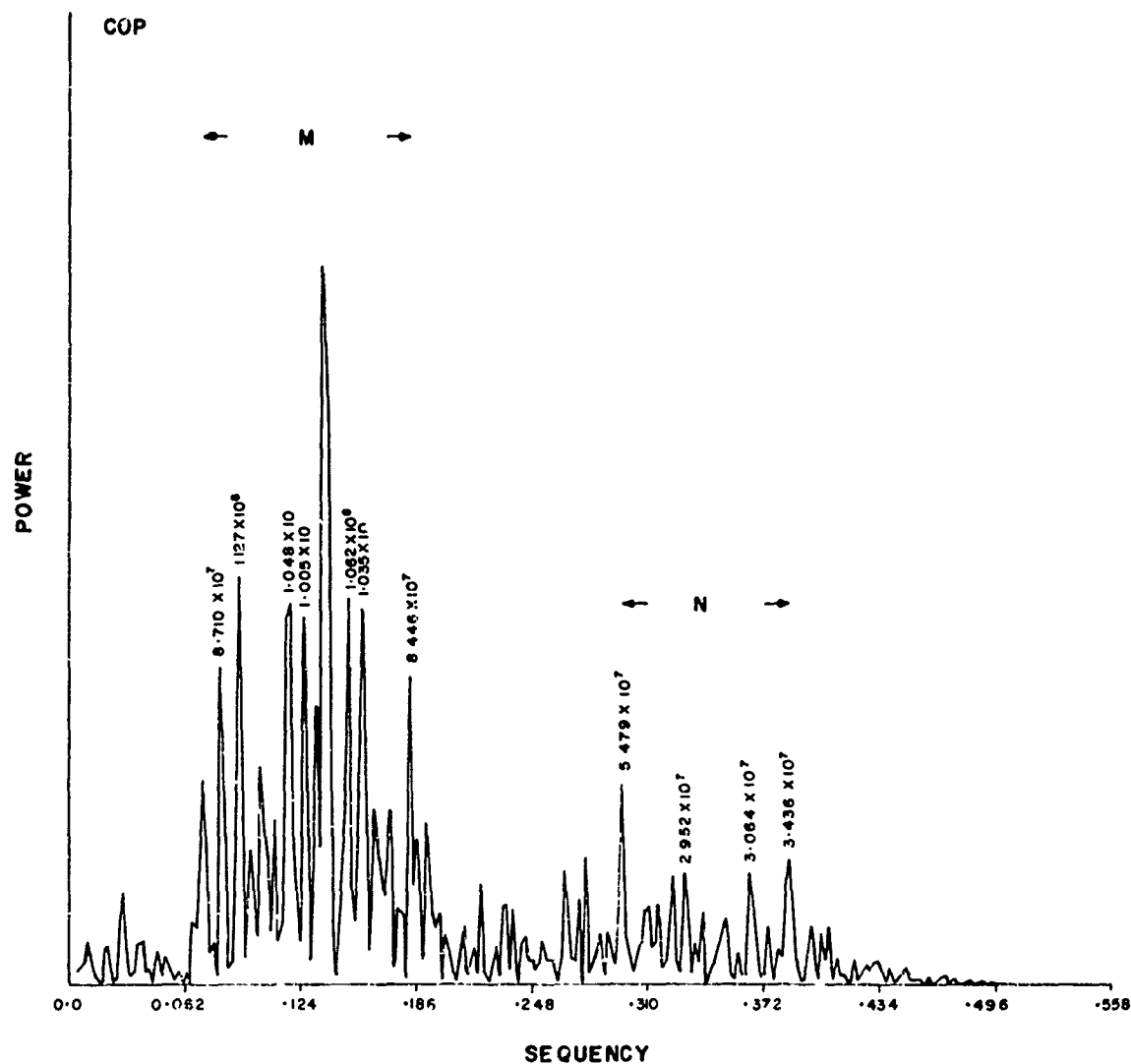


Fig. 8 Sequence power spectrum recorded at Copenhagen.

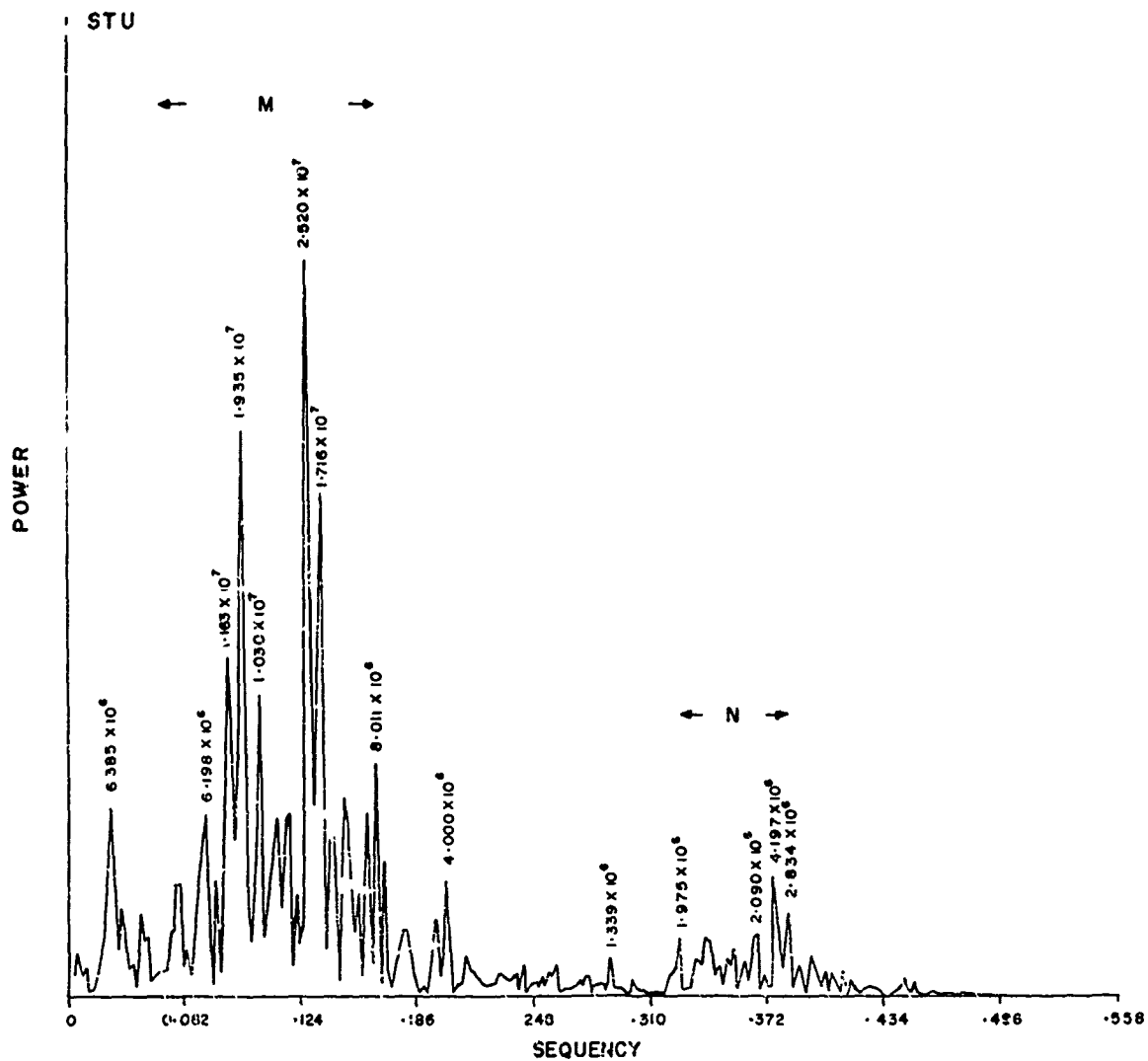


Fig. 9 Sequency power spectrum recorded at Stuttgart.

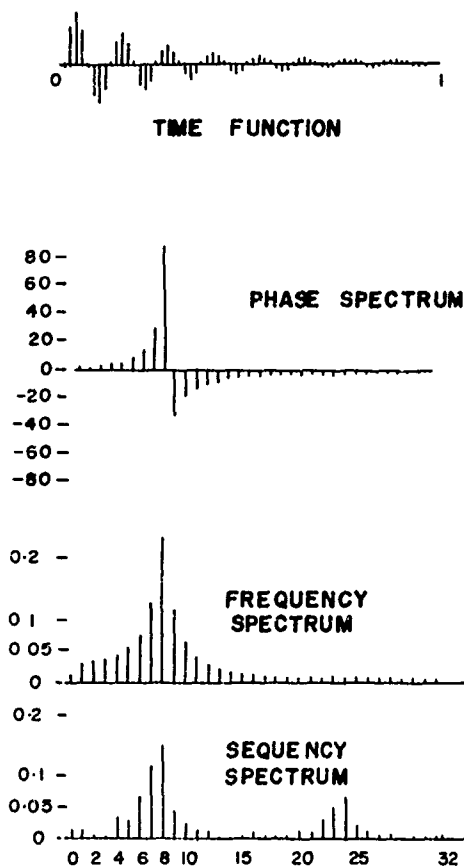


Fig. 10 Damped sinusoidal function, its phase and frequency amplitude spectrum, and its sequence amplitude spectrum.

WALSH DOMAIN PROCESSING OF MARINE SEISMIC DATA

Chi-hau Chen

Southeastern Massachusetts University
North Dartmouth, Massachusetts 02747

Introduction

The enormous amount of marine seismic data makes it necessary to process the data with a high speed digital computer. Fast Walsh transform (FWT) has been used along with the fast Fourier transform (FFT) in digital processing of marine seismic data. The digitized seismic data processed include two sets of seismic reflection profiler data from the Woods Hole Oceanographic Institution and one set of seismic refraction data from the Scripps Institution of Oceanography. The objectives of the study are two-fold: (1) to filter the marine seismic data in order to obtain a good estimate of the time of occurrence for the primary and the secondary reflections and the first refracted arrivals, and (2) to reconstruct the seismogram from the filtered data for a better interpretation of the ocean subbottom structure. In our preliminary study using IBM 360 Model 40 computer (presently limited to the 64K bytes in memory), FWT has demonstrated two distinct advantages over FFT: (1) FWT requires only one-fourth of the computation time as compared with FFT with the same number of data points, and (2) the simple linear Walsh filtering can be very effective. As a result, FWT performs better than FFT in reconstructing the marine seismogram. In this paper, the computational aspects of Walsh domain processing will be discussed and some preliminary computer results presented. Detailed results of Fourier domain processing have been reported in Ref. 1.

Development of the FWT Algorithm

Let y_N be the Hadamard transform of the data vector x_N with N data points. Then

$$y_N = \frac{1}{\sqrt{N}} H_N x_N \quad (1)$$

where $1/\sqrt{N}$ is a normalizing factor and H_N is the Hadamard matrix. The Walsh transform, denoted as w_N , of the vector x_N is obtained by rearranging all components of the vector y_N according to the sequency order. Eq. (1) requires $N(N-1)$ additions. If N is a power of 2, $N = 2^n$, fast algorithms have been considered. A somewhat different approach is used in this paper. Let \otimes be the Kronecker product. Rushforth [2] has shown that the Hadamard matrix, in general, can be written as

$$H_{2n} = (H_2 \otimes I_2 \otimes \cdots \otimes I_2) \\ (I_2 \otimes H_2 \otimes \cdots \otimes I_2) \cdots \\ (I_2 \otimes I_2 \otimes \cdots \otimes H_2) \quad (2)$$

for the positive integer n , where Eq. (2) has n factors. For $n = 3$,

$$H_8 = (H_2 \otimes I_2 \otimes I_2)(I_2 \otimes H_2 \otimes I_2) \\ (I_2 \otimes I_2 \otimes H_2)$$

which can be described by the flow chart in Fig. 1. Figure 1 can be implemented exactly in the same manner as the fast Fourier transform program except replacing the multiplications in FFT with additions or subtractions. The dots at stages A_1, A_2, A_3 represent the addition except that the subtraction is specified by a minus sign above a line. The results of each stage can be stored "in place" in the original vector x_N . This operation performs the Hadamard multiplication upon x_N . The resulting Hadamard transforms must be unscrambled according to the number of sign changes in the Hadamard row vectors, the so called sequency order. The inverse Walsh transform is an identical operation as the Walsh transform. A detailed listing of computer programs is described in Ref. 3. Presently on our computer facility the FWT program can perform transformation up to 4096 samples as compared with 2048 samples in using the FFT program.

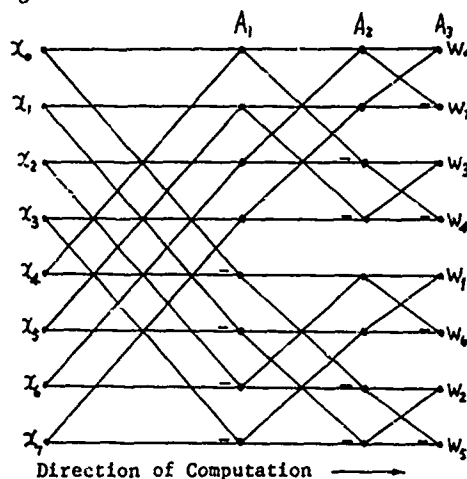


Fig. 1 Flow Chart of an Eight-Point FWT

Computer Results

The results presented in this paper are based on the second set of Woods Hole seismic data taken in Java sea, June 1971. The sampling

rate is 500 Hz. Each shot has 2048 data points including the explosion which indicates the starting time. Figure 2 is a plot of shot 522. The primary reflection which occurs at 0.33 second after the explosion provides information on the water depth. There are a number of multiple reflections which should be removed or suppressed. Walsh domain linear filtering, logexponential filtering, and nonlinear filtering as described by Pratt [4] have been applied to a number of shots. By taking the Walsh transform of shot 522 and then the inverse transform of the first half of the transformed samples, the result of the linear filtering is shown in Fig. 3. In Fig. 2, the ratio of the second peak to the first peak is 0.44. This ratio is improved to 0.406 in Fig. 3. For logexp filtering of shot 522 as shown in Fig. 4, the ratio is 0.42. Nonlinear filtering introduces many high frequency components. Nonlinear filtering of shot 522 followed by low-pass filtering is shown in Fig. 5. One way to suppress the multiple reflections is to bring each sample to a power $1 + \epsilon$, where ϵ is a small positive number, and then filter the data. For a power of 1.2, the ratio of the second to the first peak of the linearly filtered data was found to be 0.315. The disadvantage of this method is the suppression of useful but low-amplitude data. Several shots of data can be combined by simple averaging to improve the signal-to-noise ratio. Linear filtering of the average of shots 520, 521 and 522 is shown in Fig. 6. The ratio of the second to the first peak is improved to 0.307. A portion of the marine seismogram is reconstructed as shown in Fig. 7. Starting from shot 423, every three shots are combined by simple averaging and then linearly filtered. For each combined shot, the Walsh transform of the first 1024 samples is taken. This operation requires 7 seconds of computer time. The inverse transform of the first 128 transformed samples is then obtained. The filtered sample is quantized into 10 levels with the magnitude increasing from level 0 to level 9. Each vertical line of Fig. 7 corresponds to one combined shot with each filtered sample represented by a level number (level 0 is not printed). The time difference between every two adjacent samples is 15.6 msec. The computer

time for processing 60 combined shots as shown in Fig. 7 is 11 minutes and 48 seconds. This includes time for tape reading, compiling (3 minutes) and computation. To do the same job using FFT requires at least 50 minutes. Fig. 7 indicates the rise in the ocean bottom as the distance increases. The multiple reflection is particularly evident as the water depth decreases. Finally the FFT magnitude and the FWT of shot 522 are shown in Figs. 8 and 9 respectively.

Concluding Remarks

Walsh transform has provided us with an important computational method in the digital processing of marine seismic data. The improvement of FWT over FFT in both computation time and noise reduction is particularly significant in seismic study. Presently, additional Walsh filtering techniques and deconvolution using Walsh function are being examined.

Acknowledgment

The author would like to thank Dr. G. F. Anderson, Associate Director of SMU Research Foundation, for his continued support on this work and Ronald Boucher, and electrical engineering student, for his programming help.

References

1. C.H. Chen, "Digital Processing of Marine Seismic Data", report submitted to SMU Research Foundation, July 1971.
2. C.K. Rushforth, "Fast Fourier-Hadamard Decoding of Orthogonal Codes", Information and Control, Vol. 15, pp.33-37, 1967.
3. C.H. Chen, "Listing of Major Computer Programs", supplement to the report, Ref.1, October 1972.
4. W.X. Pratt, "Linear and Nonlinear Filtering in the Walsh Domain", Proceedings of 1971 Symposium on Applications of Walsh Functions, Washington, D.C., pp. 38-42, April 1971.

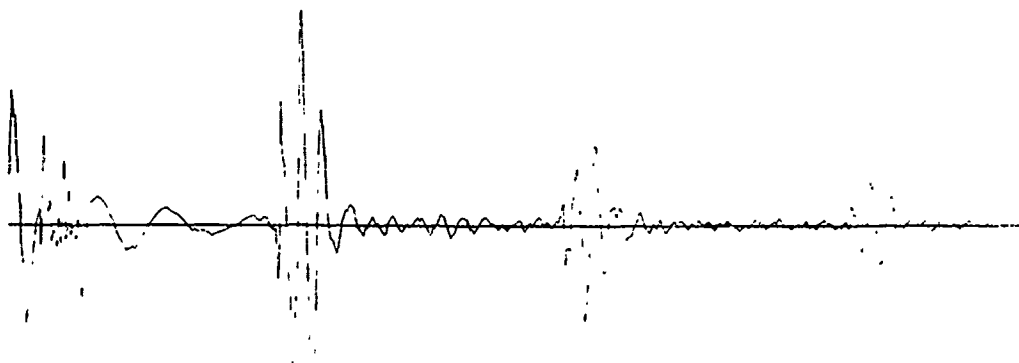


Fig. 2 A Plot of Shot 522

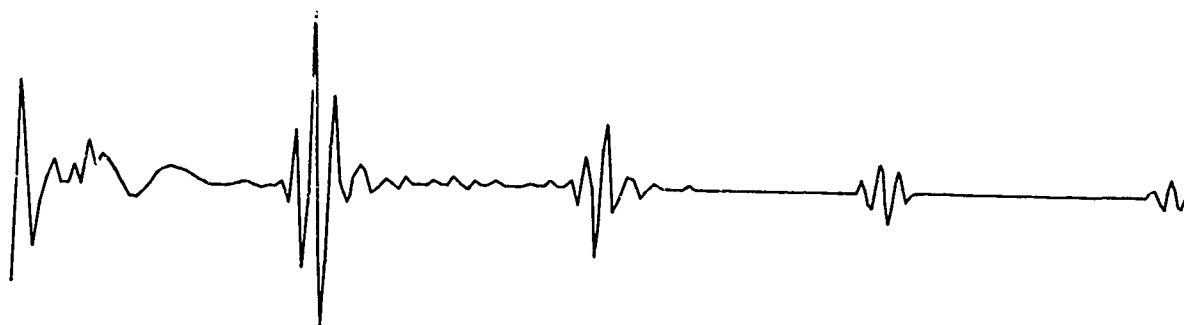


Fig. 3 Linear Filtering of Shot 522

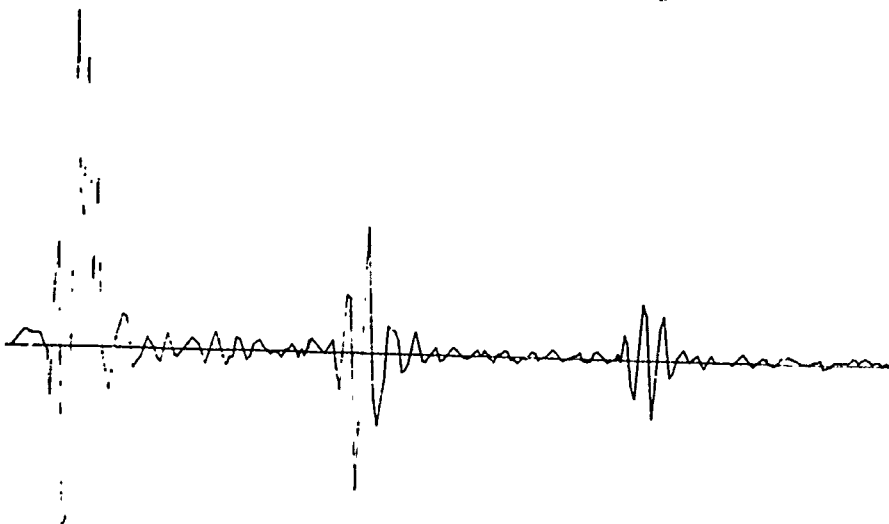


Fig. 4 Logexp
Filtering of Shot
522

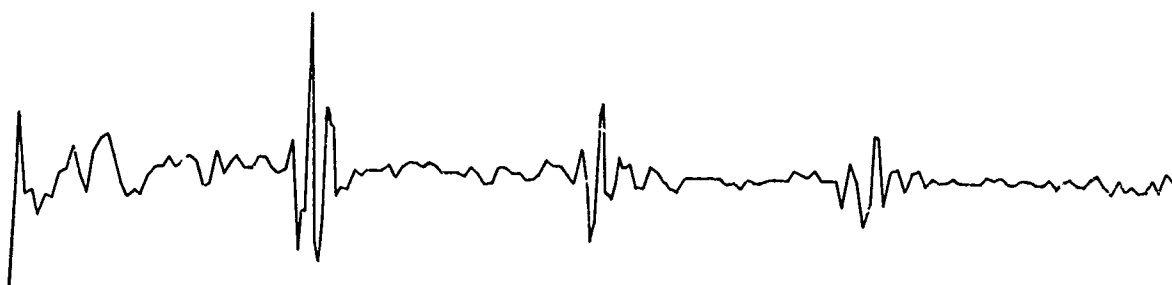


Fig. 5 Nonlinear Filtering of Shot 522

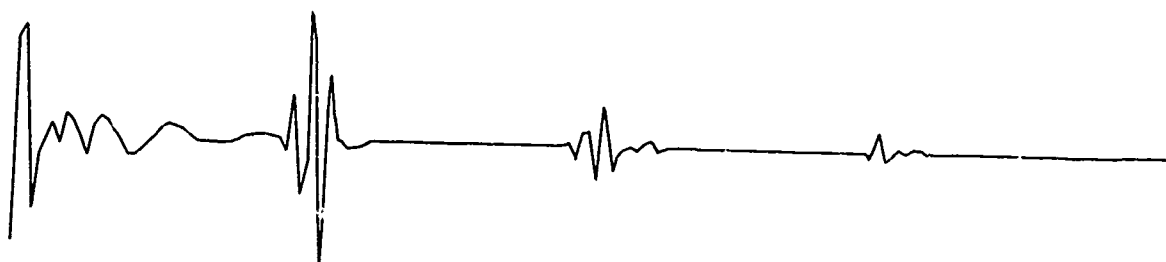


Fig. 6 Linear Filtering of the Average
of Shots 520, 521, 522

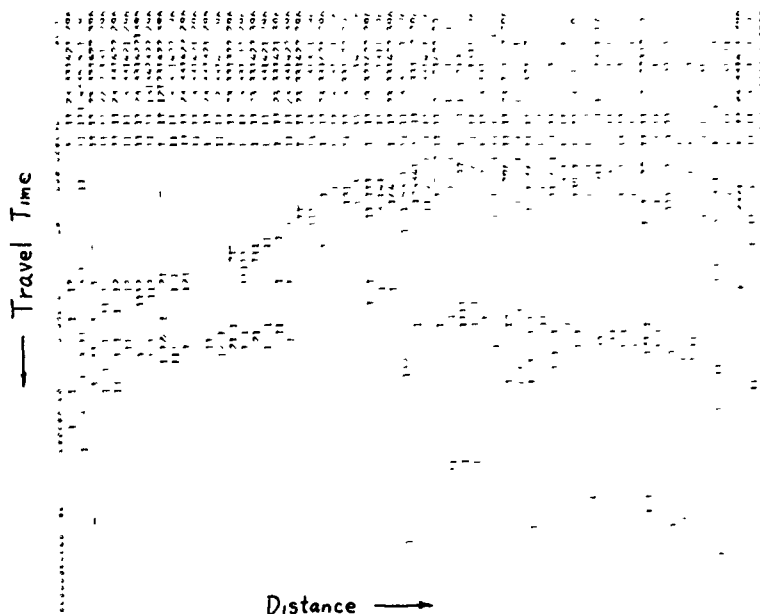


Fig. 7 Reconstructed
Marine Seismogram
(including shots
423 to 602)

Reproduced from
best available copy.

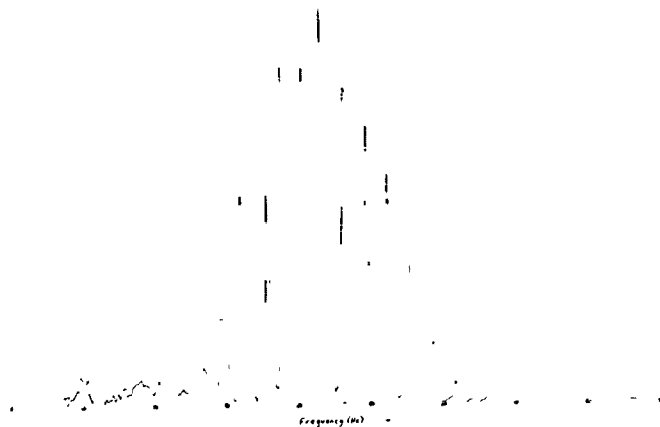


Fig. 8 Amplitude Spectrum
of Shot 522 (obtained
by using FFT)

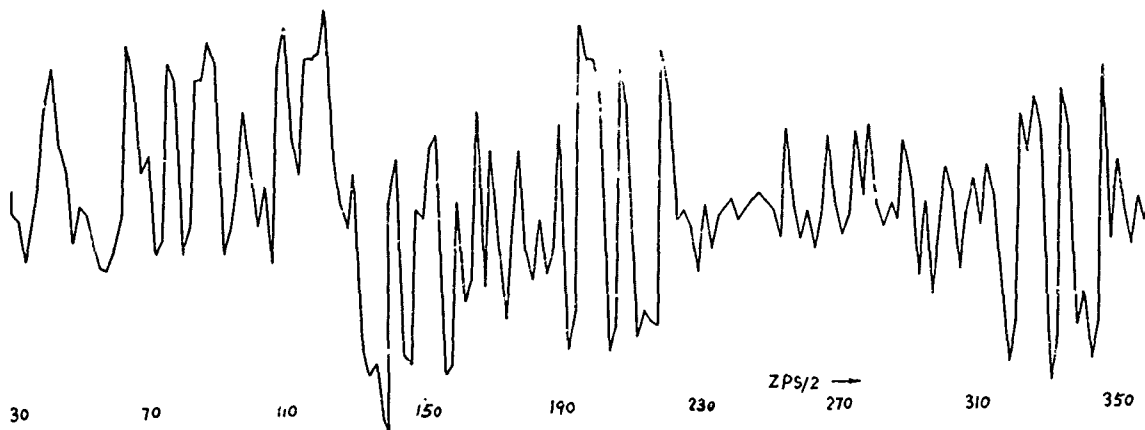


Fig. 9 A Major Portion of the Walsh Transform for Shot 522.
(plotted as a function of the number of zero-crossings per
second divided by two)

A DIGITAL INSTRUMENT FOR THE INVERSE WALSH TRANSFORM

Mr. W.O. Brown, M.Eng.
Bell-Northern Research
OTTAWA, Canada

Dr. A.R. Elliott, P.Eng.
Communications Research Lab. & the
Department of Electrical Engineering
McMaster University
HAMILTON, Canada

Abstract

This paper describes the design, implementation, and test results of a unique, inexpensive, all digital device for performing the inverse Walsh or Hadamard transform. It is suitable for operation as a programmable waveform synthesizer, and ultimately as a speech synthesizer. The inverse transform is performed using dominant term synthesis rather than using a full set of coefficient values.

Introduction

The results of various techniques for speech synthesis presented by Boswetter [1], and Campanella and Robinson [2] have shown that dominant term synthesis of speech from the Walsh domain was possible, and with fairly good quality. The orthogonal transform showed promise in reducing the bit-rate necessary for intelligible speech communication over the usual PCM technique. These preliminary results prompted this study into an instrument which would accept as an input up to eight dominant terms in either the Walsh or Hadamard domains, and then automatically resynthesize the waveform described by the coefficients. The device uses an eight-word data storage register (each word is 13 bits in length), a single "Walsh or Hadamard" function generator, one 6-bit rate multiplier, and an eight-bit binary up-down counter coupled to a final eight-bit D/A converter. It is basically a special-purpose computer, designed to be directly coupled to almost any general purpose computer, or can be controlled manually or from Read-Only memories, and designed to act as a function generator.

Theory

Being orthonormal, the Walsh and Hadamard functions can be used for a series expansion of a signal [3]. The functions are orthonormal over an interval of time θ and have the value +1 or -1. Figure 1 demonstrates the relation between the Hadamard and Walsh numbering. The Walsh series are numbered according to the "sequency" of the function. The sequency is related to the number of zero crossings within θ ; the higher the sequency, the more zero-crossings [3]. Though the instrument described here can perform either the Hadamard or Walsh inverse transform, only the Walsh series representation will be used for most of the explanations to simplify the presentation.

The Walsh Series expansion of a function $f(t)$ is defined as:

$$f(t) = \sum_{n=0}^{\infty} C(n) \text{WAL}(n, \theta) \quad (1)$$

where $C(n)$ is the coefficient of the n^{th} Walsh function $\text{WAL}(n, \theta)$. Assuming an interval of orthogonality equal to unity, any specific coefficient $C(j)$ may be determined from the following relationship.

$$C(j) = \int_0^1 f(t) \text{WAL}(j, \theta) dt \quad (2)$$

Thus the coefficients can be evaluated by equation (2), and the original waveform $f(t)$ reconstructed by applying equation (1).

For real-time analysis of signals, the integral on the right side of equation (2) can be approximated by a finite series by sampling the signal $f(t)$ a finite number of times. If the signal is sampled N times during the interval $0 < \theta < 1$ then a coefficient $C(j)$ can be approximated by the following equation:

$$C(j) \approx \frac{1}{N} \sum_{k=0}^{N-1} f_k \text{WAL}(j, \theta_k) \quad (3)$$

where f_k is the value of $f(t)$ at the k^{th} sampling instant, and $\text{WAL}(j, \theta_k)$ is the value of $\text{WAL}(j, \theta)$ at the k^{th} sampling instant. This leads to the discrete inverse transform given by

$$f(t) = \sum_{n=0}^{N-1} C(n) \text{WAL}(n, \theta) \quad (4)$$

Previous work [1,2] has shown that not all N coefficients need be used in the restructuring of $f(t)$ if a reasonable error can be tolerated. The most dominant terms tend to contain the major information for speech waveforms, in particular, dominant term synthesis is a promising means of data compression. Data compression is carried out by choosing only the M most dominant terms (in absolute value) from the complete set of N terms used in a discrete Walsh transform. The final generation algorithm then becomes:

$$f(t) \approx \sum_{n=1}^M C_{k_n} \text{WAL}(k_n, \theta) \quad (5)$$

where C_{k_n} is the n^{th} most dominant coefficient selected from a set of the first N coefficients. The value of M must be chosen, based on suitable criteria, to obtain the degree of accuracy desired, as must N .

The number N was chosen to be 64. This was a figure based on published results of several authors [1,2] which seemed to indicate that a set of 64 terms might contain the most information. (Generally, it was assumed that the information-bearing terms are the lower-ordered sequency terms). The update frequency (corresponding to the time interval θ) is a

compromise between the accuracy of the generated waveform, and the number of coefficients that must be produced in a given time interval. Because the instrument was to be used ultimately as a speech synthesizer, an update rate of 5 milliseconds was chosen as a suitable minimum update rate. Provision is made in the instrument to vary this from about 4 milliseconds to over 19 hours by simply varying the fundamental clock frequency.

The next parameter to be evaluated was M . From previous work, both at McMaster [4] and by Böswetter [1], a value of 8 for M was deemed suitable. Finally, the range of the amplitude of the coefficients was chosen. This effectively quantizes the C_k to a predetermined accuracy. The system presented here uses only the eight most dominant terms within an analyzed field of 64 Walsh coefficients. Within this set of the eight most dominant terms, the ratio of the absolute value of the most dominant to the least dominant coefficient (ignoring any zero-valued coefficients) for some basic waveforms was determined (Fig. 2). As can be seen in Figure 2 the maximum ratio was less than 50:1. Based on this analysis, and knowing that speech is reasonably well synthesized with an amplitude error of up to 2 or 3 percent, a ratio of 64 to 1 was chosen for the coefficients. This produces a 7-bit designation for the coefficient amplitude allowing for the sign bit. The coefficient of the j th term can have any integral value from -63 to +63.

Design

In order to perform the mathematics of equation (5), the final system must be capable of handling programmable Walsh function inputs, multiplying these by their coefficients, and then adding all the terms together. Assuming that a programmable Walsh function generator exists, the system shown in Fig. 3 will generate the required Walsh series reconstruction. Each of the eight Walsh functions are multiplied by their coefficients, and the results summed in a binary adder. The sum is converted to analog form by a final D/A converter. The process is repeated each time any of the Walsh functions or coefficients changes value. The coefficients C_{kn} and the Walsh number k_n are usually periodically updated by the driving computer.

This system is not very efficient in terms of hardware complexity, in that eight distinct Walsh function generators are required. A multiplexing system using one Walsh function generator, and one multiplier to sequentially produce the required terms for the sum is simpler. However, the binary adder now needs a memory element in it to store the partial sums as each additional term is being calculated. Since the absolute value of the Walsh function is always unity, instead of the Walsh function being multiplied by its coefficient before adding, the Walsh function value (± 1) and the sign of the coefficient can be logically com-

bined. If the Walsh function and the coefficient have the same sign, then the absolute value of the coefficient can be added directly to the partial sum in the binary adder. If the signs are different, the absolute value of the coefficient should be subtracted from the partial sum in the adder. Figure 4 is likely circuit for this implementation.

However, the complexity of the adder circuitry has increased substantially from the first system. The circuit presented in Fig. 5 is the novel circuit actually used to produce a simpler system than that of Fig. 4. The technique used converts the absolute value of the coefficients into a series of pulses. For instance, if C_{kn} equals 37, then thirty-seven pulses would be produced. These pulses are then simply counted in a binary up-down counter. The direction of the count is determined by the Walsh function generator, and the sign of the coefficient. By using such a system, the complex adder-subtractor with memory is replaced by a binary up-down counter, which inherently contains the required memory.

Operation

A block diagram of the final apparatus is given in Fig. 6. The circuit that converts the absolute value of the coefficients into a string of pulses is the rate-multiplier [5]. Basically, the rate multiplier is a device having a clock input, a series of r control lines, and an output. A binary number (C_k) is placed on the control lines, and a clock signal of frequency f_{in} is placed on the clock input. The output consists of a series of pulses having a frequency f_{out} such that:

$$f_{out} = f_{in} \cdot (a/b) \quad (6)$$

where a is the binary number on the r control lines, and $b = 2^r$.

The Walsh function generator used is one proposed by Siemens and Kitai [6], and modified to also generate Hadamard functions.

The eight input storage registers are loaded sequentially, either directly from a computer, or manually through the use of panel switches. A multiplexer under external control is used to determine which register is being loaded with the data. In this manner, data can be loaded independently while the instrument is performing the inverse transform from the previous set of data.

In order to facilitate the handling of positive and negative numbers, it was necessary to have the digital-to-analog converter centered in its range for a zero input signal. This was done by presetting the binary up-down counter to half-scale. Internally, an extra 4-bit counter was coupled to the basic 8-bit up-down counter to allow expansion to a 10-bit D/A converter if required, and to act as a scratch pad for intermediate calculations in case of overflow. Overflow does not occur, however,

if scaling of the initial coefficient values is such that the sum of the absolute value of all the coefficients' amplitude used is less than 127.

The input word for each storage register is organized as shown in Fig. 7. The 13-bit word required for each dominant coefficient includes a six-bit portion for the Walsh function number, and a 7-bit portion for the coefficient amplitude (one bit being a sign bit).

Results

The final instrument is shown in Fig. 8, while typical outputs are shown in Fig. 9. The outputs in Fig. 9 were taken directly from the D/A converter, and shown the obvious computation frequency. The signals synthesized were periodic, and hence there is no change when the coefficient values are updated every 5 milliseconds. For non-periodic waves, a change should also appear corresponding to a frequency of about 200 Hz. Filters should prevent much of this frequency component computation frequency component (12.8 KHz.) from appearing at the final output.

An error analysis shows that errors occur mainly in two ways. First, there are mathematical errors arising from using a truncated series, from quantizing the original signal before obtaining the coefficient values, and from quantizing the coefficient values to 7 bits. Secondly, there are machine errors, namely in the D/A converter, the finite transfer time from the up-down counter to the D/A converter, the calculation time (basically 1/64 of the time θ) before a new value is presented to the D/A converter, and the reset time of the counter before performing the next calculation. Machine errors (in the amplitude of the output) are a maximum of 0.8%, while quantizing of the coefficient amplitudes produces an error of about 1.6% maximum. The main source of error appears to be in the finite length of the series, and the coefficient truncation.

The error in the sine wave shown in Fig. 9 is about 2%, while the discontinuous ramp and triangle produced errors of less than 1%. The pulse waveforms have about 0.2% error, mainly due to machine limitations.

Waveform	Most Dom. Coeff. (Abs. Value)	Least Dom. Non-zero Coeff. (Abs. Value)	Ratio
Sine	63	1.3	48.5
Triangle	48	3	16
Ramp	32	1	32
Speech u	36	0	3.6

Fig. 2: Ratio of Walsh Coefficients for Some Waveforms

Conclusions

The instrument described above has been built, and is presently being interfaced to a PDP-8/L computer for continuing research in speech and waveform synthesis under computer control. The final system has a maximum frequency response of about 8,000 Hz., and a minimum of 1.5×10^{-5} Hz.

Its practical uses include a low-frequency function generator (especially for ramps, triangles, and pulses); or as a special purpose function generator, which can be programmed with a read-only memory; or (as a long term goal) a useful speech synthesizer for coupling to a computer.

References

1. C. B. Bopp, "Analog Sequence Analysis and Synthesis of Voice Signals", Proc. Symp. Appl. Walsh Functions, Washington, 1970, pp. 220-229.
2. S.J. Campanella, G.S. Robinson, "Digital Sequence Decomposition of Voice Signals", Proc. Symp. Appl. Walsh Functions, Washington, 1970, pp. 230-237.
3. H.F. Harmuth, "Transmission of Information by Orthogonal Functions", Springer-Verlag, New York/Heidelberg/Berlin, 1970.
4. Y.Y. Shum, A.R. Elliott, "Speech Analysis and Synthesis Using the Hadamard Transform", Proc. Symp. Appl. Walsh Functions, 1972, Washington, D.C.
5. A.R. Elliott, "A High Speed Binary Rate Multiplier", IEEE, Vol. 59, No. 8, pp. 1256-1258, Aug., 1971.
6. R. Kitai, K. Siemens, "A Hazard-Free Walsh Function Generator", Trans. Inst. and Measurement, Feb. 1972, pp. 80-83.

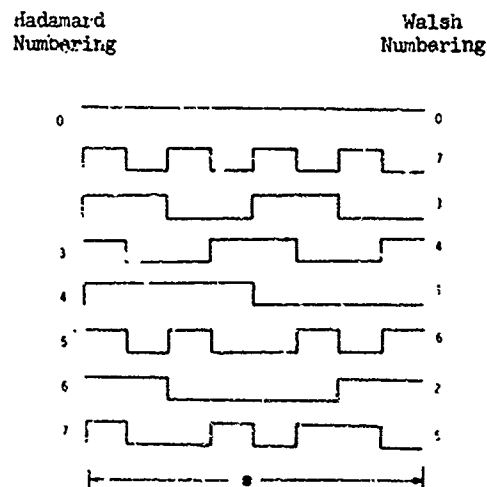


Fig. 1: Walsh-Hadamard Relation & Ordering

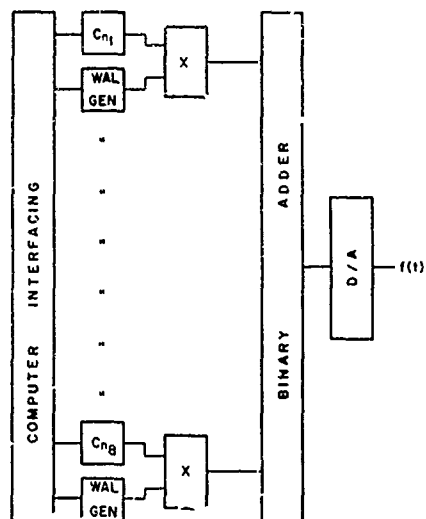


Fig. 3: Basic Instrument Design

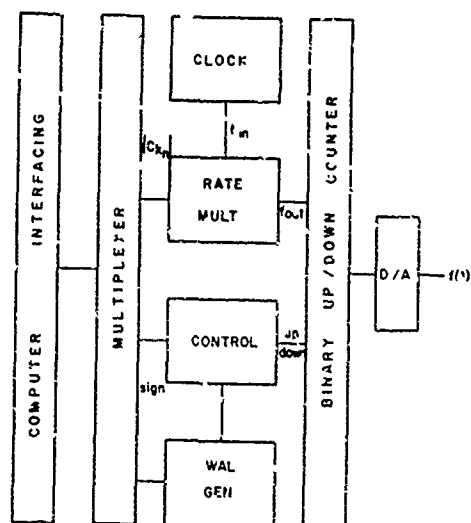


Fig. 5: Final Circuit Design

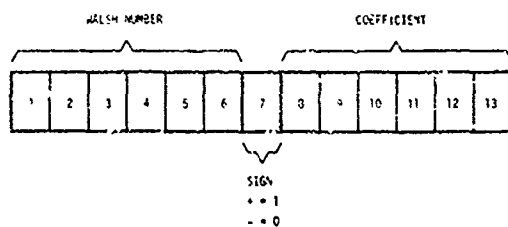


Fig. 7: Word Organization

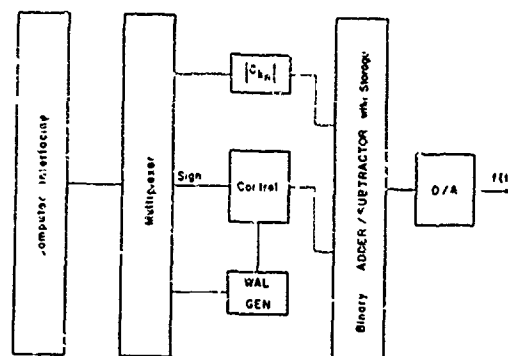


Fig. 4: Multiplexed Design

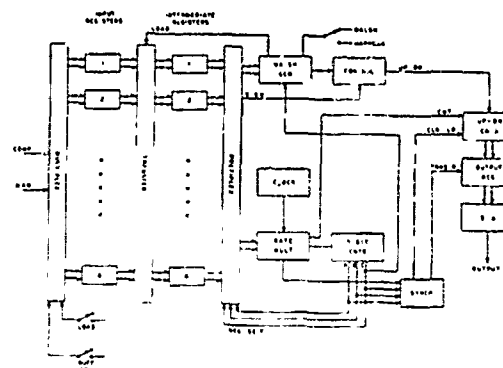


Fig. 6: Block Diagram of Inverse Transform Apparatus

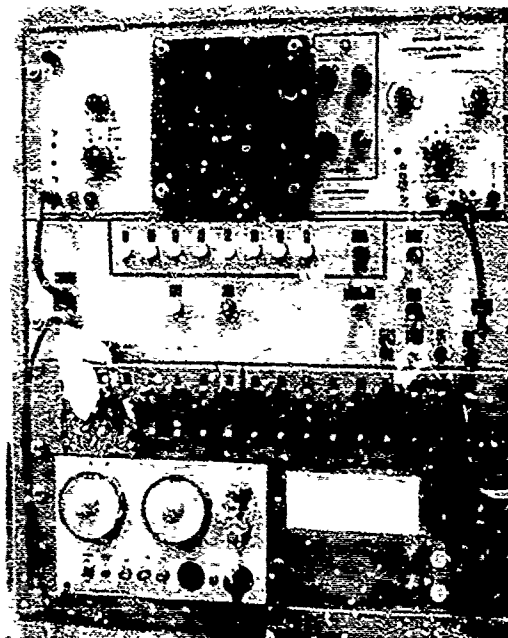
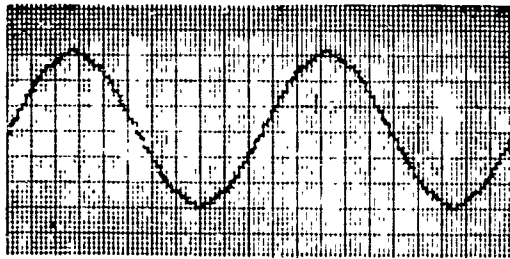
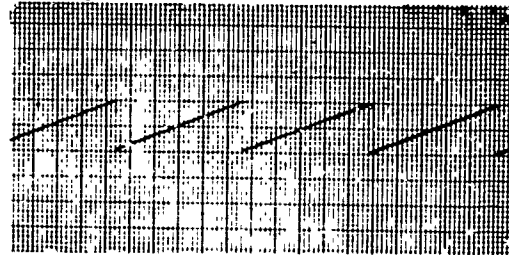


Fig. 8: Apparatus for Inverse Transform

Sine Wave: 7 most dominant terms

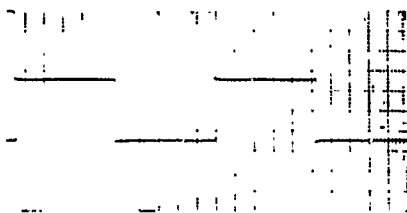


Ramp: 5 most dominant terms

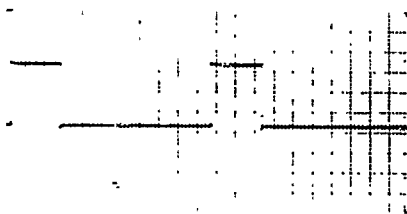


Pulse Trains

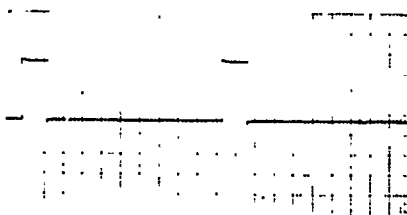
A. Duty Ratio = 0.5



B. Duty Ratio = 0.25

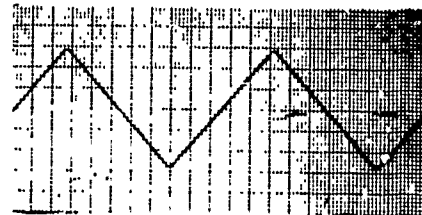


C. Duty Ratio = 0.125

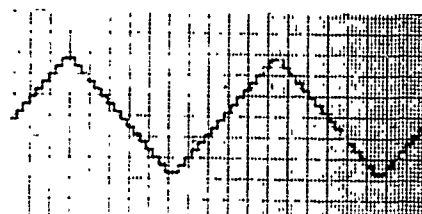


Triangle Generation

A. Five most dominant terms



B. Four most dominant terms



C. Three most dominant terms

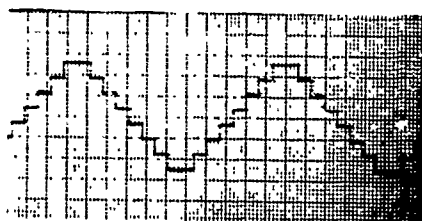


Fig. 9: Typical outputs from the Inverse Transform Apparatus

GROUP MULTIPLEXING BY CONCATENATION OF NON-LINEAR CODE DIVISION SYSTEMS

by

J.A. Gordon, M.A.,
R. Barrett, B.Sc., Ph.D, C.Eng., F.I.E.E.
The Hatfield Polytechnic,
Hatfield, Hertfordshire, U.K.

Summary

Further work on a system of multiplexing in which binary-modulated Walsh functions are summed and limited, individual channel data being recovered by correlation, is presented. In particular, a method of extending the system is indicated, in which multiplexers are concatenated.

Introduction

The normal method of multiplexing digital signals, time division, is inefficient if the load factor is variable. This is because when some channels are not in use, capacity is being wasted. Enquiries indicate that typical mean load factors (mean/peak) are in the region of 20 to 35 percent. Since most systems are designed for peak capacity, this represents a significant wastage. Recently a system has been described in which the spare capacity is used to provide constructive redundancy in such a way as to offer error-protection when the system is not fully loaded, and in particular to provide a tradeoff between the number of channels in use and the degree of error-protection offered. This system, called Correlation-Recovered Adaptive Majority Multiplexing (CRAMM), is extensively discussed elsewhere (Refs. 1,2,3) and only a few details will be included here, since the purpose of this paper is to present some theoretical results for a concatenated version.

A diagram of the system is shown in fig. 1. A set of Walsh carriers are modulated with binary data in a set of modulo-2 gates. The Walsh carriers are in logical 1-0 form with 0 corresponding to the +1 and 1 to the -1 of the normal form. A modulo-2 gate is thus equivalent to a multiplier. These modulated Walsh functions are then passed to a majority gate which must be of the type which always forms a majority, no matter how many inputs are present.

The output signal consists of the output from this majority gate plus a synchronising signal, which overwrites one of the timeslots of the Walsh carriers.

The timeslots are then re-ordered in a circuit called the distributor. This takes place just before transmission and has the effect of breaking up bursts of errors. At the demultiplexer, a corresponding redistributor

re-allocates the original timeslots to the characters. The nett result of passing through the distribution and redistribution circuits is a delay. The demultiplexer - proper consists of a shift register with normal and complementary outputs, and a set of majority gates, one for each channel. The j th input to the i th majority gate is connected to the normal output of the j th register stage if the j th character of the i th Walsh function of the carrier set is a logical 1, and to the complementary output otherwise. Register stages are measured from the back.

The outputs of these gates are then held in bistable circuits which read the majority gates at the instant the received frame fills the register, as determined by the synchronising circuitry.

Theoretical and experimental results for the system are reproduced in fig. 2. An analysis of the system is presented in the appendix. For the present we note that the number of channels which may be used depends upon the existence of a matrix, the rows of which display a sign-invariant correlation-coefficient after being modulated, summed and threshold-limited. Such a matrix is the set of Walsh functions $Wal_1(t)$ to $Wal_7(t)$ with the first character in each suppressed, these have been called Truncated Walsh functions by the authors. Computer searches indicate, and Gough has shown (4) that such matrices of Walsh functions do not exist for order 16 by 16 or greater. The original system (refs 1,2,3) used a 7 by 7 matrix, and a typical example of a signal passing through the system is indicated in fig. 3.

Increasing the number of channels

The obvious way of extending the system, namely by using a larger matrix is thus not available. Other methods which have been considered include time sharing two or more such systems TDM-wise in a longer wordlength, which will of course extend the number of channels but will not increase the error-protection.

It would seem a pity to have a system whose chief feature is its ability to adapt to a reduced load by increasing the error-protection, if this property could not be extended when the system is extended.

The only remaining method of interest is thus concatenation, for which the output of one multiplexer is taken to be the input of a second multiplexer and so on. Thus each successive demultiplexer 'sees' a lower error rate, and itself reduces this rate.

Figure 4 shows the general idea, and the notation employed.

Numbering is arranged in the conceptual order in which a signal passes through the multiplexer, with the reverse numbering at the demultiplexer, such that a demultiplexing stage with the same numbering as a given multiplexing stage performs the inverse operation to the latter. The first level of multiplexing is performed with the multiplexers numbers $(1,1), (1,2), \dots, (1,n)$; the final stage of demultiplexing is carried out with demultiplexers similarly numbered, and referred to herein as the first level of demultiplexing, performing as it does the inverse operation to the first level of multiplexing. In a similar way the second level of multiplexing is numbered 2. The k th input of the j th multiplexer at the first level of this system is labelled $1,j,k$. We term this mode of connection of multiplexers a concatenation.

In the transmission path a device to introduce independent errors is indicated, since as will be explained, we consider only the probability of a given character being in error.

We will use the terms 'error rate' and 'probability that a given character is in error' as synonymous, as implied by Bernoulli's hypothesis that the probability may be inferred by the average rate. We assume that all characters have the same probability of being in error of p , independently of all other errors, this in turn implies that any tendency towards bursts of errors has been countered by for example, reordering the characters of the signal in the time domain before transmission, and performing the inverse operation after transmission. After the addition of errors, the error rate is p with $0 \leq p \leq 1$. It is shown (in the appendix and refs. 1, 2 and 3) that for small error rates, there exists a simple relationship $f(p)$ between the signal error rate p and the channel error rate P where P is the probability that there will be a discrepancy between the input character into the multiplexer on a given channel in a given timeslot, and the corresponding character from the demultiplexer. Thus $P = f(p)$

Let the output error rate in the stream from the demultiplexer labelled 2 (second level of demultiplexer in the sense described earlier) be given by $p' = f_2(p)$. Then this error rate is the input error rate into the next (or first) level of demultiplexing.

We have assumed that signal errors are independent, and thus blocks of characters may be taken as independent with regard to the probability of obtaining a susceptible error pattern. These blocks correspond to characters for each of the demultiplexers at the first level, and thus statistical independence is maintained at the first level. Let the relationship between input error rate p and output error rate p' for the first level of demultiplexing be given by $p' = f_1(p)$, then we may say that

$$p' = f_1(f_2(p))$$

Now the relationships f_1 and f_2 are of the form

$$f(p) = kp^x \text{ and we note that}$$

$$f_1(f_2(p)) \neq f_2(f_1(p)).$$

unless the k 's and the x 's are related.

Now the k 's and x 's are functions of the number of channels in use at one time, and thus if all the multiplexers are identical, and all the demultiplexers are identical, it makes a difference to the overall error performance whether n_1 channels are multiplexed at each first level multiplexer of which there are n_2 , or vice versa. An example will make the foregoing clear. Suppose that both multiplexer and demultiplexer at both levels are of the 7 input type as has been described elsewhere. Let the first level of multiplexing be carried out with all seven channels in use and the second using three such multiplexed inputs. The input/output error relationship for a fully loaded 7 channel system is $f(p) = 2.2p$, and for a 7 channel system loaded with three channels $f(p) = 7.5p^2$. These results are derived in the appendix. Thus we have $f_1(p) = 2.2p$ and $f_2(p) = 7.5p^2$ which gives:

$$f_1(f_2(p)) = 2.2(7.5p^2) = 16.4p^2$$

If however the first level had used only three channels per multiplexer and the second level had used 7 inputs, the same number (namely 21) channels would have been accommodated but we would have $f_1(p) = 7.5p^2$ and $f_2(p) = 2.2p$ giving:

$$f_1(f_2(p)) = 7.5(2.2p)^2 = 35.9p^2$$

which is about twice the previous rate.

$$\text{If we let } f_1(p) = k_1 p^{x_1} \text{ and } f_2(p) = k_2 p^{x_2}$$

then we have

$$f_1(f_2(p)) = k_1 (k_2 p^{x_2})^{x_1} = k_1 k_2^{x_1} p^{x_1 x_2}$$

$$\text{and } f_2(f_1(p)) = k_2(k_1 p^{x_1})^{x_2} = k_1^{x_2} k_2 p^{x_1 x_2}$$

and we see that in general these are not equal since unless either:

$$k_1 = k_2 \text{ or } x_1 = x_2 = 1 \text{ we have}$$

$$k_1 k_2^{x_1} \neq k_1^{x_2} k_2$$

These results are summarised in table 1 for the 7 channel system. In this table the rows refer to the first level and the columns to the second level. In the first row and column are included the TDM case, for reference purposes.

The error relationship for TDM is $f(p) = \text{unity}$, thus if one of the multiplexers is TDM it is as if there were no concatenation.

The same results are presented graphically in fig. 5.

The Authors would like to thank the Director, Signals Research and Development Establishment, under whose sponsorship the work was carried out as a research agreement with the Ministry of Defence (Procurement Executive).

Table 1
Error rates for
concatenated system

2nd 1st	TDM	7 chans	3 chans	1 chan
TDM	p	2.19p	7.5p ²	35p ⁴
7 channels	2.19p	2.19(2.19p)	2.19(7.5p ²)	2.19(35p ⁴)
3 channels	7.5p ²	7.5(2.19p) ²	7.5(7.5p ²) ²	7.5(35p ⁴) ²
1 channel	35p ⁴	35(2.19p) ⁴	35(7.5p ²) ⁴	35(35p ⁴) ⁴
2nd 1st	TDM	7 chans	3 chans	1 chan
TDM	p	2.19p	7.5p ²	35p ⁴
7 channels	2.19p	4.78p	16.4p ²	76.6p ⁴
3 channels	7.5p ²	35.9p ²	421.9p ⁴	9187.5p ⁸
1 channel	35p ⁴	801.4p ⁴	110 742.2p ⁸	52 521 875.0p ¹⁶

APPENDIX

We consider a set of code words \bar{A} such that a_{ij} is the j th character in the i th codeword. The number of characters in each codeword is L and there are n codewords. Thus \bar{A} is an n by L matrix. Let a_{ij} have the two possible values ± 1 . Let \bar{D} be a column vector of n components each of which may have one of the two values ± 1 . \bar{D} will be called the data vector. The vector $\bar{B} = \bar{D}^t \bar{A}$ has as its j th component the sum of the j th components of the codewords, each weighted by the data appropriate to it. We define the operator 'signs' to have the effect of replacing a vector by another vector whose components are the signs of those of the original vector.

The signal transmitted by the C.M.A. system is thus:

$$\bar{S} = \text{signs}(\bar{B}) = \text{signs}(\bar{D}^t \bar{A})$$

The demultiplexer finds the correlation coefficients of this signal with each of the codewords. The column vector of such correlation coefficients is given by

$$\bar{P} = \bar{A} \bar{S}^t = \bar{A} (\text{signs}(\bar{D}^t \bar{A}))^t$$

and the column vector of recovered data is given by the signs of these correlation coefficients:

$$\bar{D}' = \text{signs}(\bar{A} (\text{signs}(\bar{D}^t \bar{A}))^t)$$

It will be observed that the notation has been chosen such that events which take place serially such as the transmitted signal, are represented by row vectors, and simultaneous events, such as the input of data, by column vectors, thus in all these matrices, 'time' goes horizontally.

We define the column vector \bar{R} to be

$$\bar{R} = \bar{D}_{\text{diag}} \bar{A} \text{signs}(\bar{D}^t \bar{A})$$

where \bar{D}_{diag} has as its i th component the data d_i , and zeros everywhere else. Thus if r_i is positive, the recovered data on the i th channel is the same as that at the input, and there is no error. \bar{R} has been called the 'reassurance vector' and r_i the reassurance on the i th channel. If one character in the signal \bar{S} is altered, the reassurances on each channel will change by ± 2 .

Thus a reassurance of $2t+1$ is a guarantee of protection against t errors.

Let n_p be the number of characters in a signal \bar{S} which contribute positively to r_i and

n_n be the number which contribute negatively.

To begin with we have $n_p + n_n = L$

and in addition, by the choice of the partition of the characters we have

$$n_p - n_n = r_i$$

Thus we get

$$n_p = \frac{1}{2}(r_i + L)$$

We now consider two cases depending on whether L is even or odd, (actually in the case to which most attention has been given (1,2,3) $L = 7$ is odd).

L Odd

The addition of an error to one of the n_p positively contributing characters will reduce n_p and increase n_n by one, and thus reduce r_i by two. We may add errors until r_i is reduced to a negative number, in which case a channel error will result. The number of errors which may be so added is $\frac{1}{2}(r_i - 1)$, and the addition of one more, namely $\frac{1}{2}(r_i + 1)$ errors, could lead to a channel error providing all the signal errors occur among the n_p positively contributing characters. We say that the channel is susceptible to

$$x = \frac{1}{2}(r_i + 1) \text{ errors.}$$

On the assumption that the most likely course of channel errors is that which requires the least number of signal errors, namely when exactly x signal errors occur in a signal such that the i th channel is susceptible to x errors, we may say that the probability that exactly x errors will occur among the n_p positively contributing characters is

$$\binom{n_p}{x} p^x (1-p)^{n_p-x} \approx \binom{n_p}{x} p^x$$

and the probability that an arbitrary word is susceptible to x errors is $1/2^n$ on the given channel. Then the most likely source of errors is a data vector of least susceptibility. Let n_{\min} be the number of data vectors of least susceptibility x_{\min} .

The probability of error in the i th channel is given by:

$$P_i = \frac{N_{\min}}{2^n} \binom{n_{\min}}{x_{\min}} p^x$$

where $n_{\min} = \frac{1}{2}(r_{\min} + L)$ and $x_{\min} = \frac{1}{2}(r_{\min} + 1)$

and r_{\min} is the least reassurance on the i^{th} channel.

L even

Here the number of errors to reduce r_i to zero is $\frac{1}{2}r_i$ provided they occur in the n_p positively contributing characters. We assume that the majority gate will be designed to round upwards or downwards for 50-50 divisions, and so half of these cases will lead to error.

Otherwise the argument is identical to the case for n odd.

$$P = \frac{N_{\min}}{2^{n+1}} \binom{n_{\min}}{x} p^x$$

with $n_{\min} = \frac{1}{2}(r_{\min} + L)$, $x_{\min} = \frac{1}{2}(r_{\min} + 1)$

and r_{\min} is the least reassurance on the i^{th} channel.

In order to evaluate these formulae it is necessary to have a table of reassurances for all input data vectors. This information is best provided by computation and the nett result is that for the three cases considered (one, three and seven channels in use) we obtain the following results:

Number of channels in use	Channel error rate
7	$2.19p$
3	$7.5p^2$
1	$35p^4$

References

1. Gordon, J.A., and Barrett, R. 'Correlation-Recovered Adaptive Majority Multiplexing', Proc.IEE, vol.118 nos 3/4 Mar/Apr 1971 pp417-422.
2. Gordon, J.A., and Barrett, R., 'Digital majority Multiplexing using Walsh Functions', Washington, April 1971.
3. Gordon, J.A., and Barrett, R., 'On Coordinate Transformations and Digital Majority Multiplexing', Symposium on Walsh Functions, Hatfield Polytechnic, June, 1971.
4. Gough, I.R., Private communication, to be published.

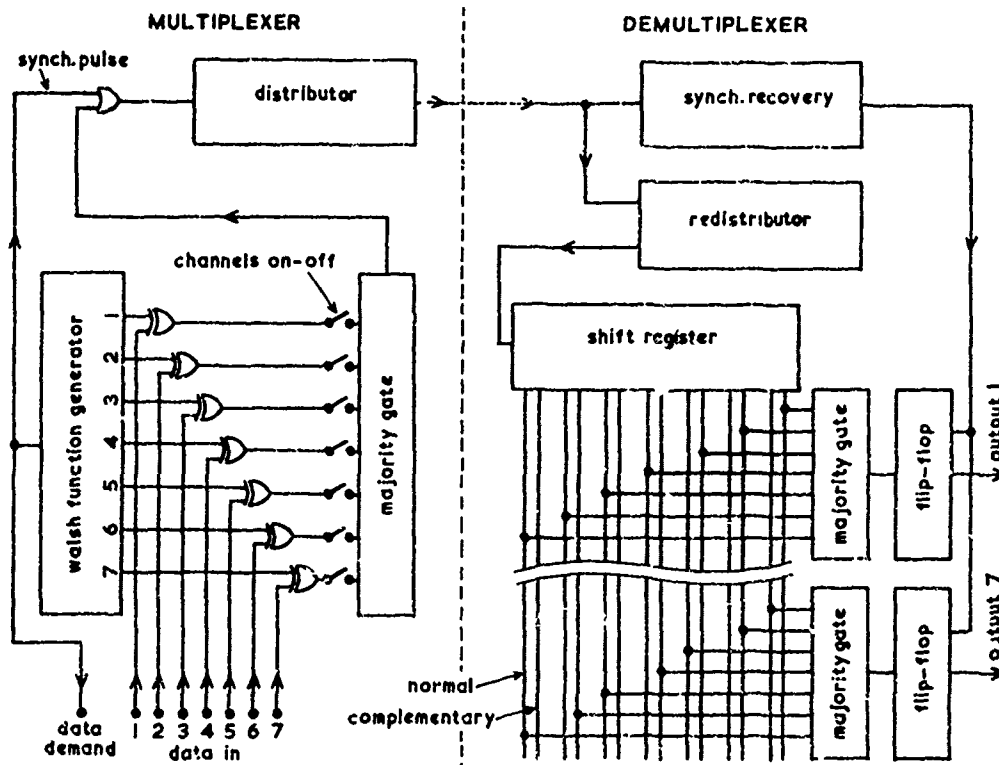


fig. 1

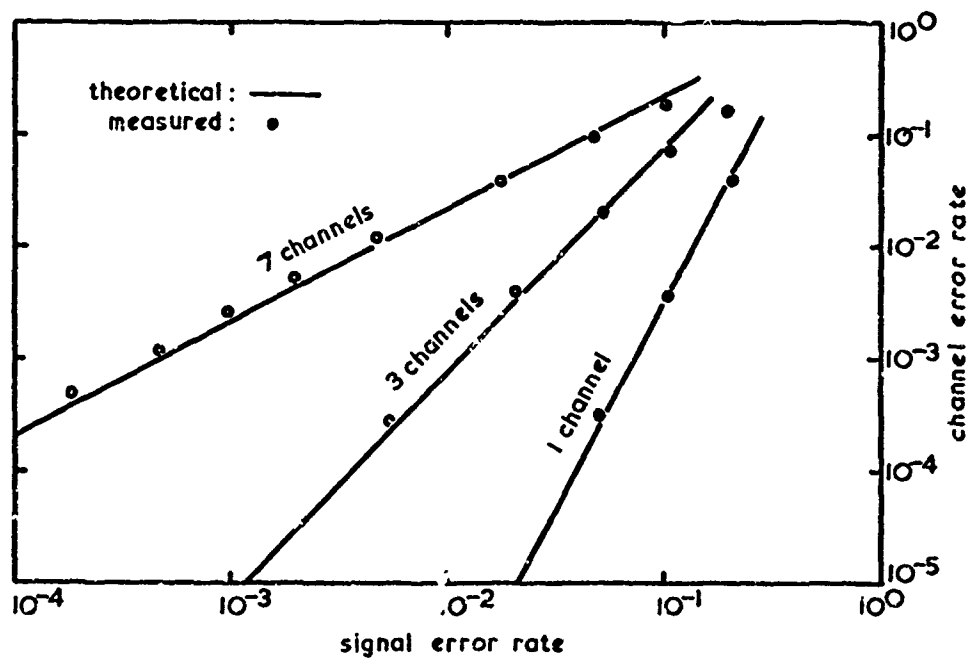


fig. 2

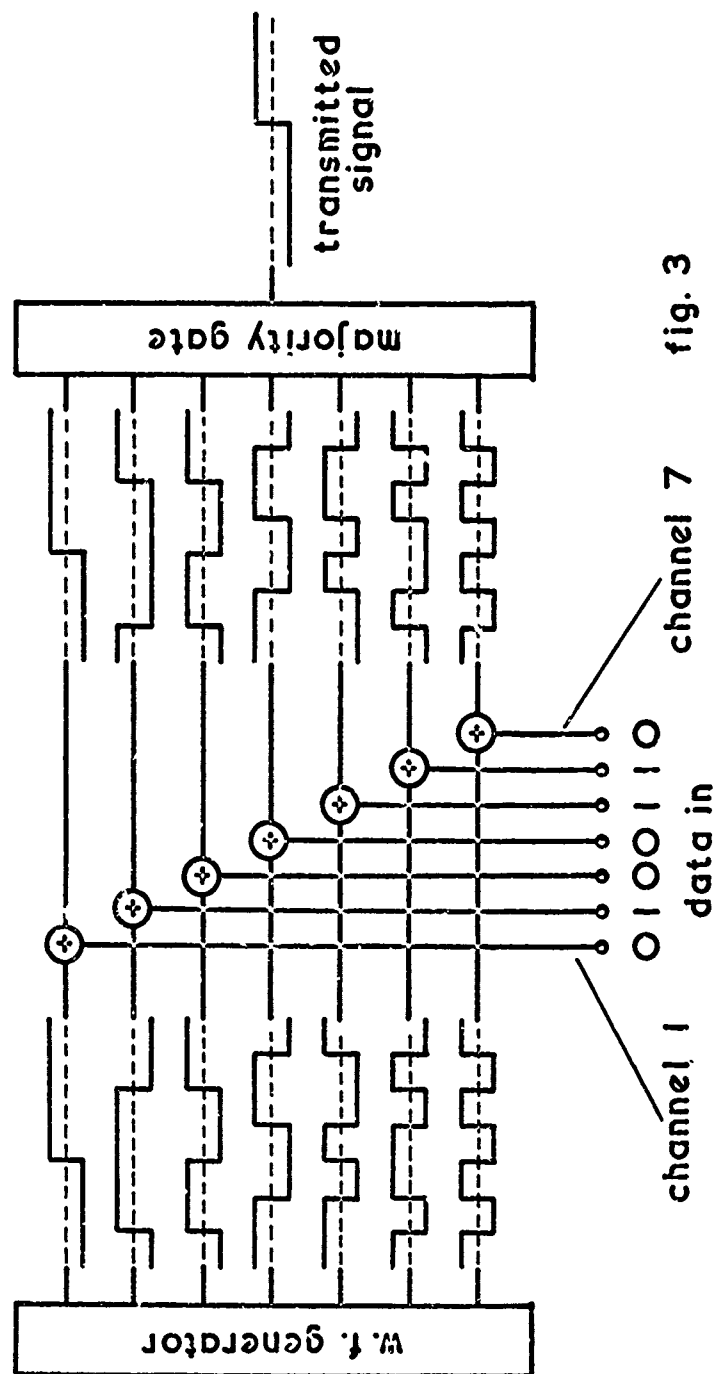


fig. 3

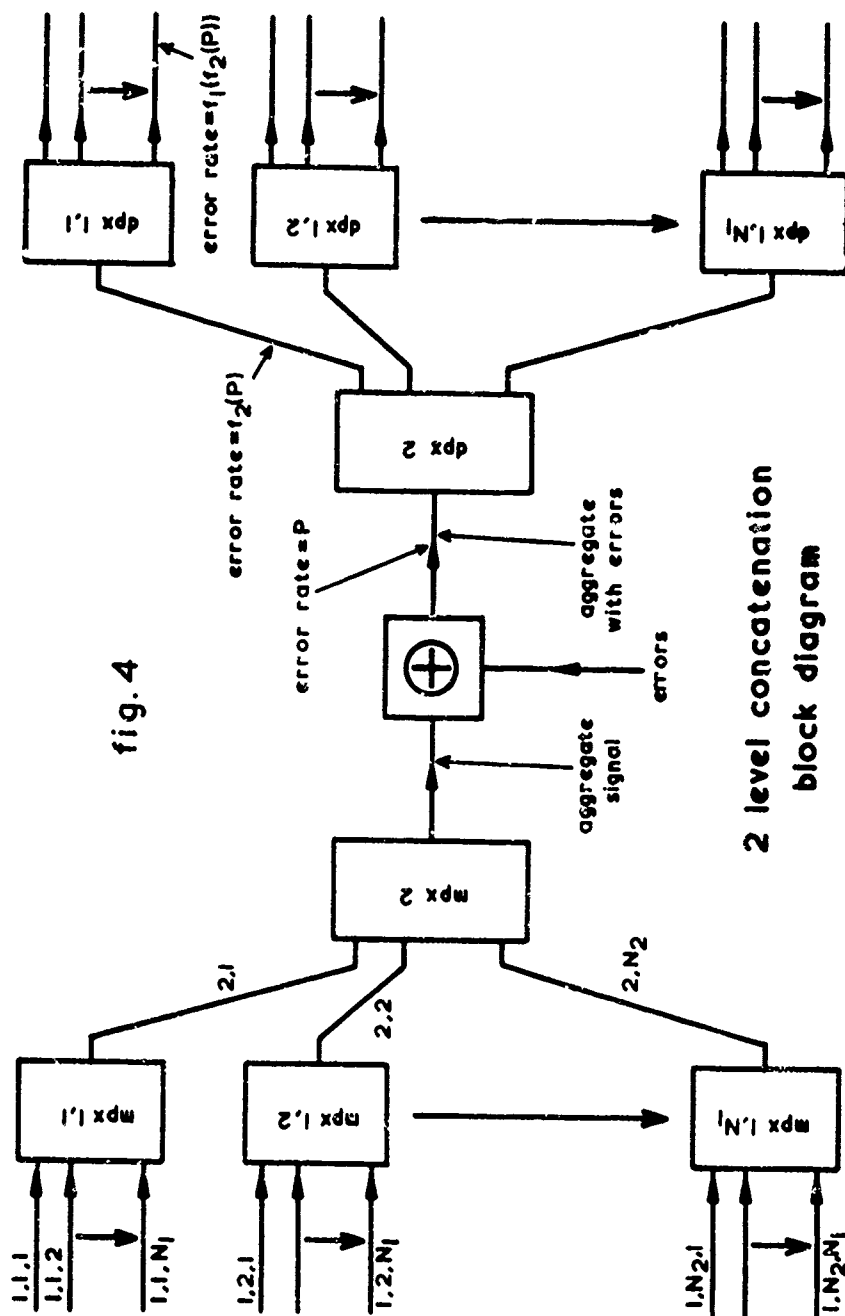


fig. 4

2 level concatenation
block diagram

error rates for 2-level concatenation

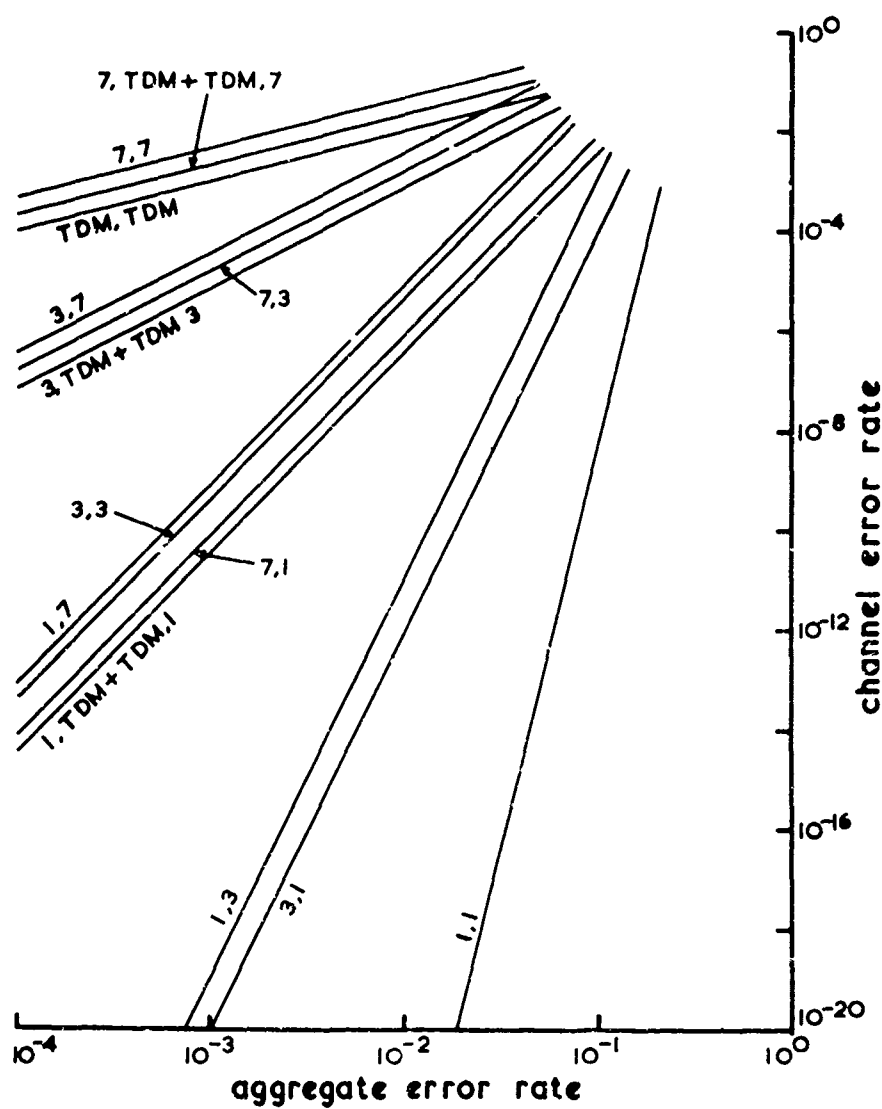


fig. 5

RESULTS OF MULTIPLEXING EXPERIMENTS USING WALSH FUNCTIONS

by David L. Durst
Grumman Aerospace Corporation

Abstract

An investigation was conducted to study the use of Walsh functions as orthogonal carriers in both analog and digital multiplexers for several of Grumman's electronic development projects. An analog multiplexer, relying heavily on the use of integrated circuits, was constructed and tested, revealing encouraging simplicity of implementation, while pointing out various circuit deficiencies which restrict system performance. A digital system, based upon an adaptive majority multiplexing scheme, was simulated on a digital computer. The data which was obtained concerning the distribution of sequence power can be of value in influencing the choice of Walsh carriers to be used in this system and gives insight into the nature of certain deterministic errors involved in system operation.

Introduction

The Grumman Aerospace Corporation is active in a number of areas of electronic development which could benefit from the use of Walsh functions. These include: interior multiplexing of data and control signals on board aircraft where cable weight is a significant factor, high-speed computer pro-

cessing of transforms, the enhancement and compression of surveillance photographs, and navigational ranging using time-shift dependent spectra. Multiplex systems are our greatest interest at this time. This paper deals with an evaluation of a prototype Walsh function analog multiplex system, and results of a computer simulation of an all digital, adaptive majority multiplexing technique.

Analog Multiplexer

The analog system whose block diagram is shown in Figure 1, is capable of modulating up to 16 band-limited signal channels onto a set of Walsh carriers for transmission over one line, and reconstructing the signals at the receiving end. While only two channels have been activated and tested, complete 16-channel operation can be achieved by the addition of several redundant analog circuit boards. The multiplexer was designed and built for evaluation, study, and demonstrations. Photographs of the complete system are shown in Figure 2.

The system was constructed entirely of readily available digital and linear integrated circuits. The Walsh function generator, consisting of a clock oscillator, binary counter and exclusive-or logic gates,

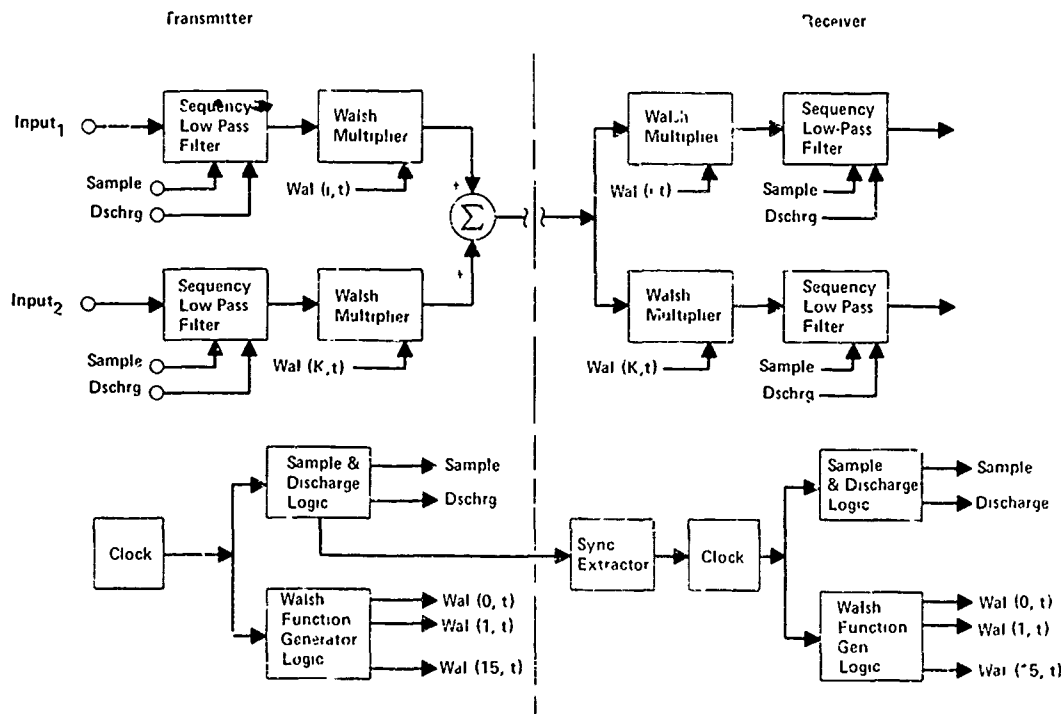


Fig. 1 Walsh Function Analog Multiplexer

was fabricated from standard 7400 series TTL digital logic. The analog processing is completely performed by LM201 operational amplifiers and DG141 analog switches. The system clock frequency was selected to produce a 150 microsecond orthogonality interval for the Walsh functions, thus providing a theoretical input bandwidth for the system of approximately 3 KHz (usable for intelligible voice transmission). Some oscillograms of representative system waveforms are shown in Figure 3.

It is of interest to note that by pushing the 7400 series of digital logic to its maximum speed capability, it is possible to generate the 16 low-order Walsh functions with an orthogonality interval

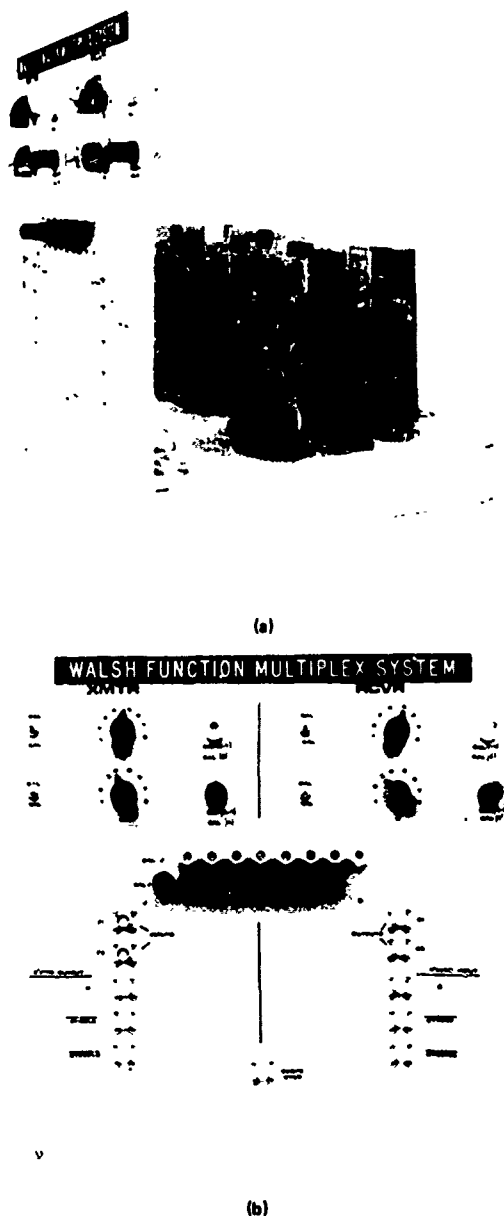


Fig. 2 Photographs of Prototype Analog System

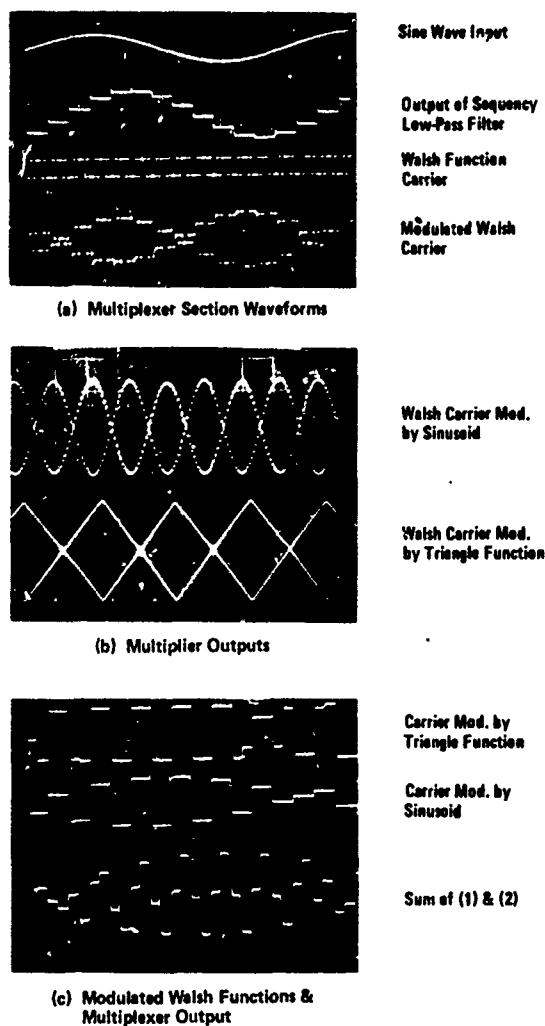


Fig. 3 Oscillograms of Representative System Waveforms

as small as 0.5 microseconds, or a maximum sequence of 30 Mega zps. Neglecting analog circuit limitations, the maximum frequency bandwidth of the input signals could be as high as 1 MHz. Furthermore, using state-of-the-art, emitter-coupled integrated logic circuits, such as the Motorola MECL III family, sixteenth-order Walsh functions can readily be generated with orthogonality intervals as low as 25 nanoseconds. This corresponds to a maximum input frequency bandwidth of 20 MHz. The linear integrated circuits which are available for our analog circuit requirements severely limit the operating sequence of this multiplexer. This is described later.

Performance Data

An important performance parameter in a multiplex system is crosstalk between channels. Measurements were made to determine the magnitude of the crosstalk in this system for all possible combinations of 16 Walsh function carriers. It was

found that the degree of crosstalk is strongly dependent upon the sequence of the Walsh function carrier selected for each channel, with the magnitude varying between -13 db and -45 db. Figure 4 shows how the crosstalk in a channel increased as the sequence of the adjacent channel carrier was increased. An increase in the amplitude of the input analog signals was also found to increase the crosstalk, although a change in signal frequency had no effect, thus eliminating the possibility that capacitive coupling between channels was involved.

The total harmonic distortion of the output signals introduced by the multiplexer was found to be a function of the sequence of the Walsh function carrier selected. As shown in Figure 5, the percent harmonic distortion of a 100Hz input signal was found to vary from .8% for a wal (0,0) modulated carrier to 2% for the wal (15,0) carrier. All distortion measurements were made with a 3-kHz single-pole low-pass filter in cascade with the output to eliminate the high frequency sampling harmonics inherent in the multiplexer.

Synchronization

Due to the nature of the correlation function of the Walsh functions, precise phase synchronization between the multiplexer and demultiplexer is essential to proper system operation. In order to observe the effects of phase shift in the prototype multiplexer, we provided an external synchronization link capable of varying the phase between the two Walsh function generators in precisely controlled increments. Figure 6a shows a graph of the amplitude gain through one channel of the multiplex system as a function of the phase shift between the cal (7,0) carriers. A similar graph for the sal (1,0) carrier is shown in Figure 6b. Each measurement was made with a step change in phase of $1/64$ th of the orthogonality interval. The graph is representative of the absolute magnitude of the autocorrelation function of the carrier and agrees with theoretical results. Increases in crosstalk, with phase shift, of up to 10 db were also observed.

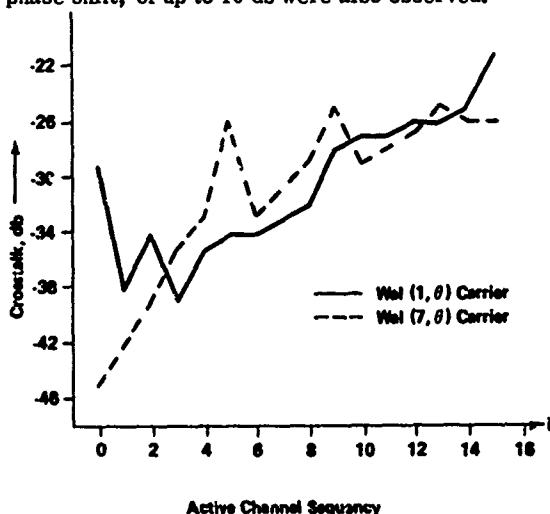


Fig. 4 Crosstalk in Designated Channel as a Function of Adjacent Channel Sequence

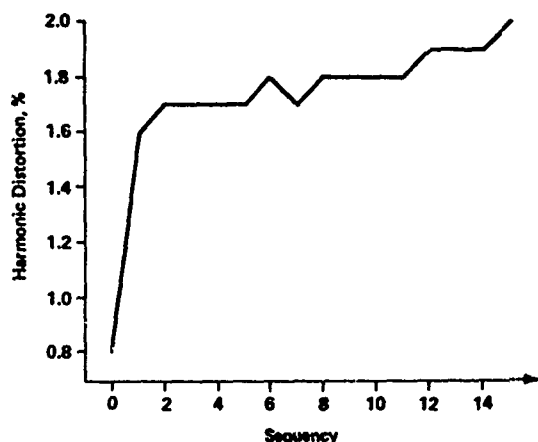


Fig. 5 Percent Harmonic Distribution of 100 Hz Sinusoid as a Function of Walsh Carrier Sequence

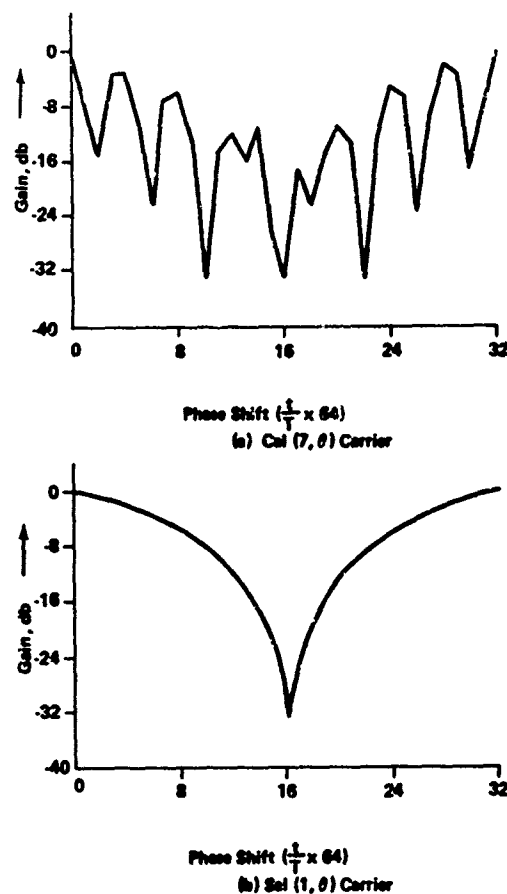


Fig. 6 Multiplexer System Gain as a Function of Walsh Generator Phase Shift Between Transmitter and Receiver

For higher frequency Walsh functions, a very slight change in phase can result in a significant change in channel output. As an example consider the $\text{wal}(7, \theta)$ function with an orthogonality interval of 1 microsecond. Figure 6a, shows that a synchronization error of as little as 16 nanoseconds (1/64 period) can result in a 20 db change in received signal output. As the frequency of the carriers which this multiplexer is designed to handle, increase, precise control of time delays through the synchronization circuits becomes critical.

Analog Circuit Restrictions

The degree to which the modulated frequency carriers within the multiplexer circuits approach perfect Walsh functions is an important factor in explaining previously mentioned deficiencies in system operation. Although good quality Walsh functions can be synthesized digitally, the ability to perform multiplication and addition operations with these functions requires linear devices which can slew between voltages of opposite polarity in zero time. Such impractical devices would be optimized in the frequency domain. In order to approach this optimum device with practical hardware, the properties of frequency-domain-optimized linear amplifiers can be combined with time-domain-optimized electronic switches. The multiplier configuration shown in Figure 7 is an example of this. This circuit which was used in our prototype system multiplies the $\text{wal}(0, \theta)$ component of the input analog signals by a selected Walsh carrier by switching the amplifier from an inverting to a non-inverting mode in step with a digital control signal.

It is desirable to use an operational amplifier in this circuit for precise gain control and frequency control characteristics. Modern integrated circuit operational amplifiers, however, are limited to slew rates of approximately 100 volts per microsecond at unity gain. Furthermore, integrated analog switches have maximum speeds of the order of one microsecond, and non-symmetrical turnon vs turnoff delays of approximately 500 nanoseconds. Because of these analog circuit limitations, the multiplexer in multiplying two near-perfect Walsh functions produces a distorted Walsh function having much longer transition times. This is a direct result of the degradation in performance involved in the interface between high-speed digital logic with limited-slew-rate linear circuitry.

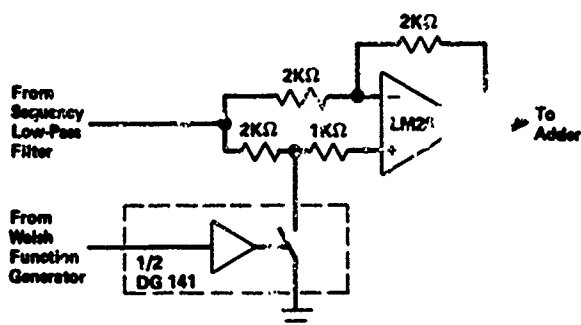


Fig. 7 Walsh Function Multiplier. Combination of Linear Amplifier and Electronic Switch

In an effort to preserve the orthogonality of the Walsh functions, it is necessary to reduce the maximum frequency, so that the finite transition times are a small fraction of the orthogonal interval. This, of course, reduces the sampling frequency of the frequency low-pass filter, thus limiting the maximum frequency bandwidth of the input signals.

As was previously mentioned, the percent harmonic distortion of the input signal channels processed by the multiplexer and the crosstalk between multiplexer channels will, in general, increase as the frequency of the carriers increase. This is in line with the frequency limitations of the Walsh multiplier and other linear circuits used in this multiplexer. Furthermore, the increase in crosstalk with increases in the amplitude of the multiplexer input signals is related to the higher voltages to which the Walsh multiplier must slew. The correspondingly longer transition times of the modulated Walsh function result in a cross-correlation function in the demodulation process which is increasingly greater than zero, thus producing higher levels of crosstalk. Amplitude compression and limiting of the multiplexer input signals will reduce the voltages to which the linear amplifiers must slew. This compromise can be used to reduce crosstalk level when faster slew-rate amplifiers are unavailable.

These experiments and observations with the prototype, have reaffirmed the feasibility of constructing a multiplex system using amplitude-modulated Walsh functions, rather than sinusoidal functions. However, certain deficiencies in available integrated circuit performance, impose various restrictions on the system design. While Walsh functions of frequency greater than 100 Megahertz can readily be synthesized using state-of-the-art digital circuits, the linear integrated circuits which can modulate these functions and preserve their orthogonal properties are severely speed limited in comparison.

Further improvements in linear integrated circuits will certainly enhance the performance possibilities of Walsh function multiplexers. It should be noted, however, that in any application where multiplication by an analog signal is involved, the inherently binary Walsh functions are immediately transformed into a multilevel analog signal, and all further processing must be done by linear circuitry. A much more desirable application of Walsh functions would retain their binary structure throughout all stages of processing, and thus make more efficient use of their properties. Such an all digital system is discussed below.

Digital Multiplexer

Barrett and Gorden [1] have described an adaptive majority multiplexing scheme in which several channels of binary data are multiplexed by modulation onto the seven lowest-order Walsh functions. Each bit of binary input data is represented in terms of the $\text{wal}(0, \theta)$ function, with a logic "zero" designated as $+\text{wal}(0, \theta)$ and logic "one" designated as $-\text{wal}(0, \theta)$. If the logic level for each input channel is multiplied by its respective Walsh carrier, and the products are algebraically added, the system would be identical to the linear system

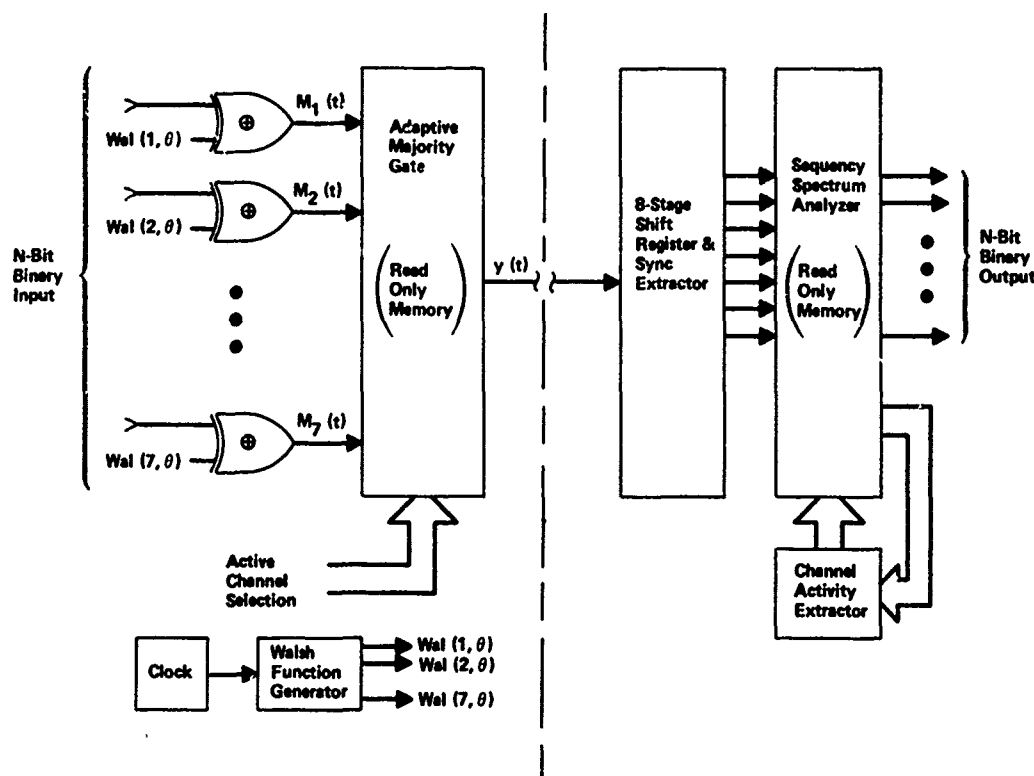


Fig. 8 Adaptive Majority Multiplex System

previously described. This linear superposition of Walsh functions can be hard-limited by a majority gate before transmission [1]. The resulting binary signal will, in most cases, still contain separable input logic information for each channel. It has been shown that such a multiplexing system is then capable of providing automatic tradeoff between its redundancy properties and the number of channels in use at one time.

The digital multiplexer shown in the block diagram of Figure 8 can be built entirely of digital integrated circuits, thus enabling high-speed operation. Exclusive OR logic gates can be used as Walsh-function multipliers and the adaptive majority gate can easily be implemented with a 256-bit read-only memory chip. In fact, it is conceivable to implement the entire multiplexer or demultiplexer circuitry with one read-only memory and a shift register.

Computer Simulation

A computer simulation of this system was performed in an effort to study an inherent deterministic error [1], and to determine the sequency power distribution of the multiplexed signal. The computer program simulated actual system operation with from one through seven channels active at any one time and with all possible input logic combinations generated and tested.

The computer was programmed to calculate the results of each step of the adaptive majority multiplexing process, and makes a bit-for-bit comparison

between the binary levels selected at the input and the resulting levels calculated at the output. If a match is found, a new word is calculated and the process is repeated until all possible binary bit combinations are exhausted. If an error is detected in any output bit channel, a warning message is printed, indicating the calculated output results containing the error, the corresponding input binary bits, and a representation of the Walsh sequency spectrum of the multiplexed signal. When all possible input combinations for a given number of active channels is exhausted, an additional channel is activated and the process repeats until all seven channels are activated and checked. The designation of the sequency of each Walsh carrier per channel is provided as input data for each run. It should be noted at this point that for synchronization purposes in a practical system, the first time slot of each orthogonal interval of the multiplexed signal is truncated and forced to a logic "one". The computer program uses this truncated signal but there is little difference in the final results whether the signals are truncated or not.

Computer Results

The system operates satisfactorily when the number of active channels is either one, two, three, six, or seven. When five channels are active, deterministic errors are generated for certain combinations of input logic levels. This is true for all possible choices of five out of seven Walsh carriers. Each of the 21 possible combinations of five carriers were tested by the computer program. Each run showed similar errors in the output binary bits. For the four-channel case it was possible to choose several combinations of four Walsh carriers such that the

multiplex system operated without any deterministic errors. A list of three sets of Walsh function carriers which can be used when only four multiplexer channels are operative is presented in Table 1. Time limitations prevented a check of all 42 possible choices of four carrier sets.

TABLE 1 THREE SETS OF WALSH FUNCTIONS, SUITABLE FOR FOUR-CHANNEL MULTIPLEXER OPERATION

Carrier Sequence (XT)	1	2	3	4	5	6	7
Set 1	X	X			X	X	
Set 2	X		X	X		X	
Set 3		X	X	X	X		

Majority Multiplexer Spectral Analysis

The computer-derived spectral analysis of the multiplexer output signal indicates a spreading and shifting of the spectrum of the signal as it passes through the majority gate. This, of course, should be expected since the majority gate effectively receives a multi-level superposition of Walsh functions at its input and performs a hard-limiting operation on it to produce a binary output. The deterministic error inherent in five-channel operation and most cases of four-channel operation of the multiplexer is due entirely to a loss or inversion of certain spectral components of the multiplexed signal as they pass through the highly non-linear majority gate. The resulting signal which is transmitted to the demultiplexer either contains erroneous information or doesn't contain any information concerning the binary state of the channel associated with the affected spectral component.

An example of this phenomenon is shown in Figure 9. Figures 9a and 9b show the majority gate input and output spectra representative of the signal generated by logic levels "0", "0", "1", "0", "0" modulating Walsh carriers wal (1, θ), wal (2, θ), wal (3, θ), wal (4, θ), wal (5, θ), respectively. The spectra for logic levels "1", "0", "1", "0", "0" modulating the same carriers are shown in Figures 9c and 9d. Note that the output spectra for the two different input signals are identical, and the wal (1, θ) component which was present at the input has disappeared completely. All the information determining the logic level that was multiplexed from channel number one for this particular set of input signals has been destroyed by the non-linear limiting process, and the multiplexed signal has become ambiguous.

Similar input and output spectra are shown in Figure 10 for input logic levels "00001" and "10001". Although both input spectra contain energy only in the frequency range wal (1, θ) through wal (5, θ), in this case the majority gate processing has transferred energy to the wal (0, θ), wal (6, θ) and wal (7, θ) sequences. Again for these selected five input channels, note that the output spectra for different inputs are identical. Yet for this particular set of inputs, no spectral component has disappeared completely. It is, of course, impossible for the demultiplexer to correctly identify the logic level transmitted on channel number one for this particular set of inputs, since the demultiplexed channel number one logic level would always be identified as

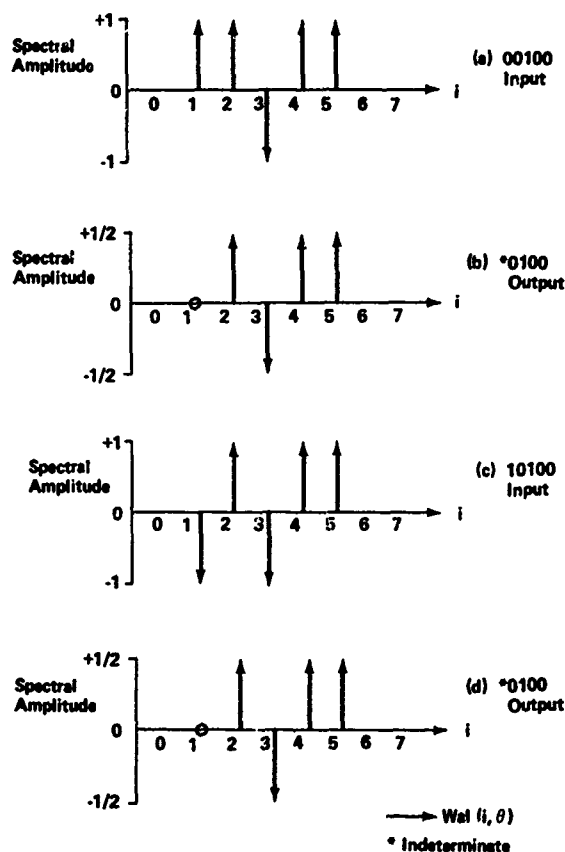


Fig. 9 Binary Input and Output Spectra for Five Input Majority Gate (Sequence of Active Carriers: 1, 2, 3, 4, 5)

"zero" regardless of the binary bit actually transmitted. In general, for each choice of five Walsh carriers, eight pairs of five-bit input word sets were found to be ambiguous (out of a total of 32 sets).

For the case of four active multiplexer channels, it was noted that only certain choices of Walsh carriers will produce unambiguous operation. Figure 11a shows the input spectral components for the four-bit input word "1100". Note that in the output spectrum of Figure 11b, the wal (4, θ) spectral component has been inverted, causing the multiplexed signal to be incorrectly identified as "1101" by the demultiplexer. A better choice of Walsh carriers for four-channel operation is shown in Figure 12a, where four input logic bits "1001" are modulated onto Walsh carriers wal (1, θ), wal (2, θ), wal (5, θ) and wal (6, θ). The resulting output spectrum in Figure 12b shows that the spectral components representing the active Walsh carriers will be correctly demodulated at the receiving end of the system. Indeed, as noted in Table 1, all 16 possible four-bit input combinations will produce valid outputs for this particular set of Walsh carriers.

Spectrum Spreading

As a result of the spectrum spreading caused by the non-linear majority gate processing, the multiplexer output signal includes all Walsh components from wal (0, θ) to wal (7, θ). It is therefore im-

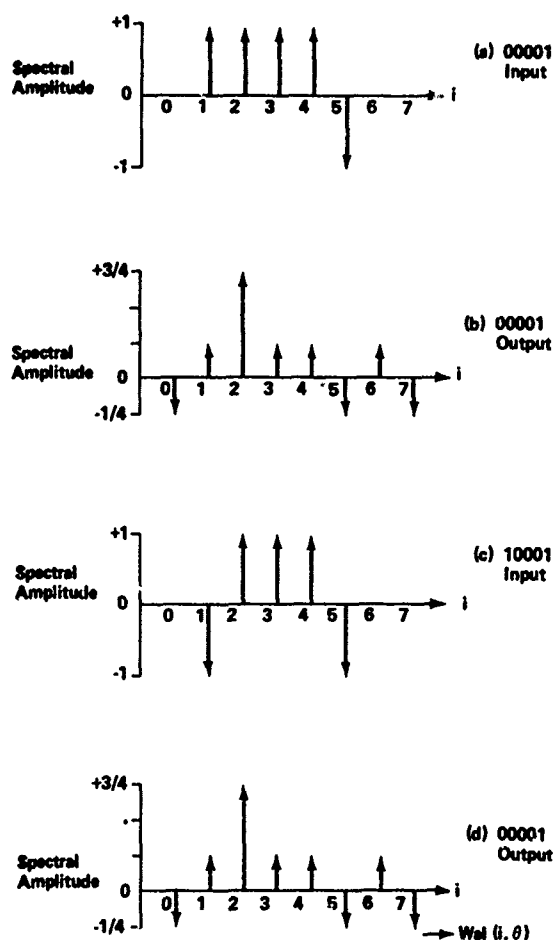


Fig. 10 Binary Input and Output Spectra for Five Input Majority Gate (Sequence of Active Carriers: 1, 2, 3, 4, 5)

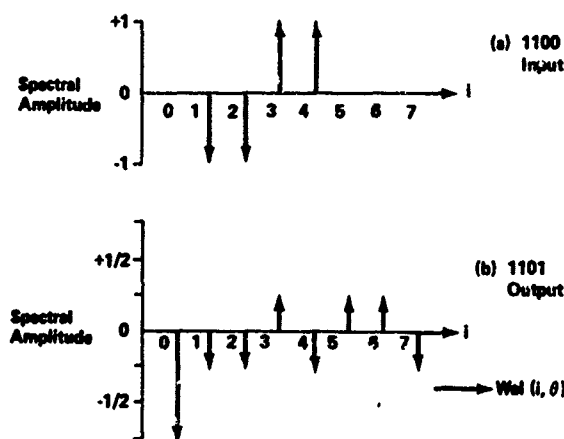


Fig. 11 Input and Output Spectra for Four-Input Majority Gate (Sequence of Active Carriers: 1, 2, 3, 4)

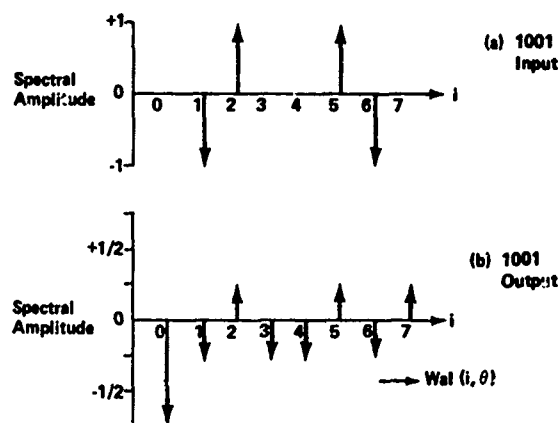


Fig. 12 Input and Output Spectra for Four Input Majority Gate (Sequence of Active Carriers: 1, 2, 5, 6)

perative that the demultiplexer "knows" which channels are active and the sequence of the carriers servicing each channel. This requires some kind of link between the multiplexer and the demultiplexer to convey channel activity information. One of the multiplexer channels could, of course, be reserved for this purpose.

Conclusions

Although the multiplexer will not operate properly when only five channels are needed, the information for one of the five channels can, of course, be redundantly multiplexed onto a sixth channel for this special case. When four channels are needed, it is essential that the carriers be selected properly. Furthermore, regardless of the number of active channels, it is essential that the demultiplexer be informed, in some manner, as to which Walsh components must be ignored. Otherwise, erroneous information will be received on inactive channels. In general, the adaptive majority technique is an attractive multiplexing scheme that can make effective use of modern digital integrated circuits, and the binary nature of Walsh functions.

Acknowledgement

The author wishes to thank Charles Schmidt, of Grumman Aerospace Corporation, for his continuous encouragement and suggestions during the course of this work, and Heinz Schreiber also of Grumman Aerospace, for stimulating discussions which contributed significantly to the preparation of this paper.

References

1. Gordon, J.A. and Barrett, R. "Digital Majority Logic-Multiplexer Using Walsh functions", Symp. Application of Walsh Functions Washington, 1971.
2. Gordon, J.A. and Barrett, R. "Correlation Recovered Adaptive Majority Multiplexer", Proc. I.E.E., Vol. 118, April, 1971.
3. Harmuth, H.F., "Transmission of Information by Orthogonal Functions", Springer-Verlag Lib. of Congress Cat. 77-118682, 1970

A WALSH-FUNCTION POWER-CABLE MONITORING SYSTEM

Frank Furrer, Arvind Shah and Martin Maurer

Institute of Applied Physics
Swiss Federal Institute of Technology
Zuerich

1. Abstract

The continuous supervising of the temperature and a number of possible other parameters of a power-cable increases its economy and its reliability considerably. It is necessary to measure the temperature at many different points along the cable and to transmit these values to a central monitoring station. This central station then possesses all the information necessary to load the cable best. The required transmission of information from many measuring points to one single central station is done by Walsh-functions.

2. Introduction

This paper deals with the application of Walsh-functions to the problem of monitoring a power-cable. The problem as defined by the cable producers, was as follows [1] :

A system which can monitor a power cable (specifically an oil pressure cable) along a length of approx. 500 ft should be designed. Information on the local surface temperature of the power-cable and a number of possible other parameters (such as oil leakage) is to be gathered from 50 measuring points distributed along the cable and transmitted to a central monitoring station (Fig. 1.).

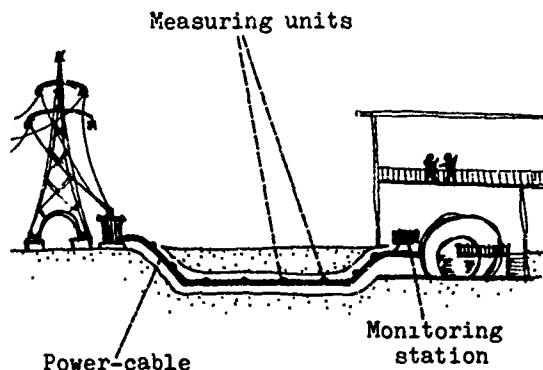


Fig. 1: Monitored power cable

Such a system would be of considerable commercial importance because it would permit a better utilization of power cables. A cable with an unknown temperature profile cannot be utilized to its proper limit because of the danger of thermal breakdown. Thus, with the monitoring system, a considerable amount in the layout of an underground power distribution network could be saved.

This problem, although inspired by the specific application of power cables, can be considered typical of many monitoring situations or alarm networks, such as might be required for pipelines, gas conducts, railway lines, etc.

It is typical of such an alarm or monitoring system that a great number of measuring or control points have to be connected over a few common wires. The use of some form of multiplexing becomes imperative: it seems that Walsh-functions are particularly suited for this power-cable monitoring situation, because the transmission rates are very low and a whole signal-cable network can be reserved entirely for the monitoring system. In fact, the higher harmonics (in the frequency domain) of the Walsh-functions do not become into conflict with the cut-off frequency of the signal-cable or with other transmission channels as is the case e.g. if frequency multiplexing is used in telephone or radio communications.

The practical advantage of frequency multiplexing for this particular application becomes evident when one looks into the further conditions that are imposed on implementation:

- The measuring elements and all local electronic circuitry required for multiplexing has to be miniaturized because they will be mounted between the lead and the thermoplastic sheath of the power cable (Fig. 2). This is necessary for monitoring the surface temperature of the lead and to detect any oil leakage. Any inductances or large capacitances required for sine-wave generation thus fall out of

consideration, and preference must be given to simple digital circuitry that can be packed into a medium scale Integrated Circuit.

- The reliability of the measuring elements and multiplexers should be typically larger than that of the power cable itself, which has a mean-time to failure of over 20 years. This again calls for Integrated Circuitry.

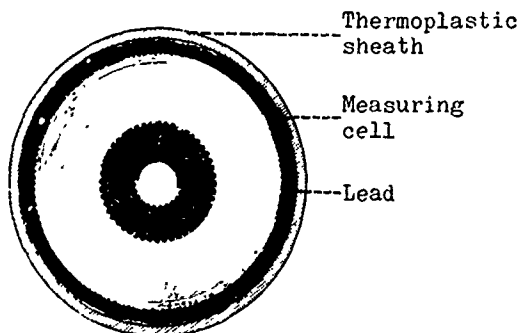


Fig. 2: Power cable cross-section

- A large amount of disturbances (noise, spikes, bursts) are expected on the transmission wires of the monitoring system: these could be due to short-circuits, load changes, etc. which are present on the high-voltage, high-power-cable. This condition calls for a system with a high noise-immunity and there again, a Walsh-function system synchronized by strong synchronization pulses from the central monitoring station has the added advantage that correlation detection at the central station is possible.

A time multiplexing technique could not be used just because the required high noise immunity would call for long transmission times for the single measuring point to the central monitoring station. This would lead to unpermissible high cycle-times for the supervision of the whole cable.

3. Description of the System

We have basically a structure with a channel, a central monitoring station and many measuring or sending cells (Fig. 3).

To every measuring cell one particular Walsh-function is assigned. This Walsh-function carries address and information of the cell at the same time.

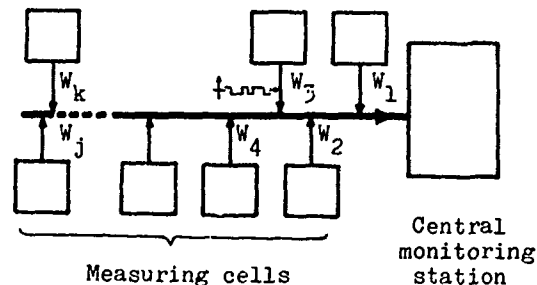


Fig. 3: Structure of the System

Each unit therefore acts as a Walsh-function generator. A common clock is fed from the monitoring station to all the peripheral units: it is used to clock a binary counter in every unit, the various outputs of this counter are fed to a logic network consisting solely of exclusive-or gates. The few variable connexions of this logic network allow the programming of the desired Walsh-function.

This scheme permits to transmit one binary decision from every measuring cell to the monitoring station according to a simple convention: the presence of a particular Walsh-function means "1", the absence means "0".

The signals of all the sending units are fed simultaneously and additively to the channel. Thus the monitoring station receives the sum of all the individual signals.

To transmit more than one binary decision from every measuring cell to the central station, a signal-word (e.g. 8 bits) is defined (Fig. 4).

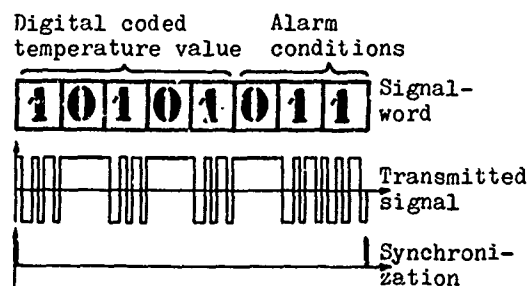


Fig. 4: Signal-word

The signal-word contains the Gray-coded information given by the temperature-sensitive device and the oil-probe. To every bit of this signal-word, exactly one period of the Walsh-function is assigned. If the corresponding bit is

"1", the period is transmitted, otherwise not.

The required synchronization is provided by a synchronization pulse at the beginning of each signal-word. The synchronization pulse is also transmitted from the central station to all the peripheral units and assures the parallelism of the signals of all the peripheral units.

In our case the channel consists of a 5-wire-system. Clock, synchronization pulses, signals and supply voltage are carried on different lines. The signals of all the peripheral units are superposed on the channel by current-addition.

At the monitoring station the signals are separated by synchronous correlation (Fig. 5). The correlation is extended over many signal words, typically 32. This is possible because the information changes very slowly: the cable has a thermal constant of hours and the occurrence of oil-alarm is unique: if oil-alarm occurs, the operation of the cable has to be stopped. As an additional feature, the information is Gray-coded, so that a change in temperature by one quantizing step always changes only one bit of the signal word. The uncertainty of the measurement arising from the "long" measuring time is therefore at most one quantizing step of the temperature value.

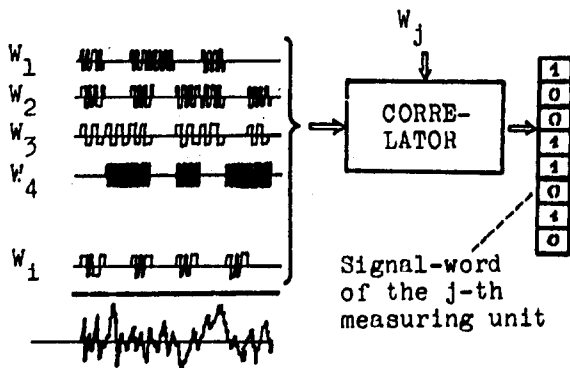


Fig. 5: Central monitoring station

This scheme allows the extraction of the data of any arbitrary sending unit, or - with the aid of a multiple or multiplexed correlator - groups of sending units. The output of the correlator is equal to the signal-word sent by the measuring cell. It can be transformed back into an analog value for display.

The time required to extract the information of all 50 units if serial calling is used is about 1 1/2 minutes (the clock-frequency in the system being 2500 Hz). With parallel

calling (multiplexed correlator) this time would be reduced to approx. 2 sec.

4. Additive Gaussian Noise

The signal arriving at the monitoring station (i.e. the Walsh-function we are interested in) has a certain signal power S , and the unavoidable noise has a power N . The signal to noise power ratio S/N at the input of the detection process is therefore given.

If we assume for the moment that only one Walsh-function is present ($s(t)=0$) and that we have ideally low-pass filtered Gaussian noise (Fig. 5), the signal to noise power ratio S'/N' at the output of the detection process, or better, the signal to noise ratio improvement

$$\eta = \frac{S'/N'}{S/N}$$

can be calculated.

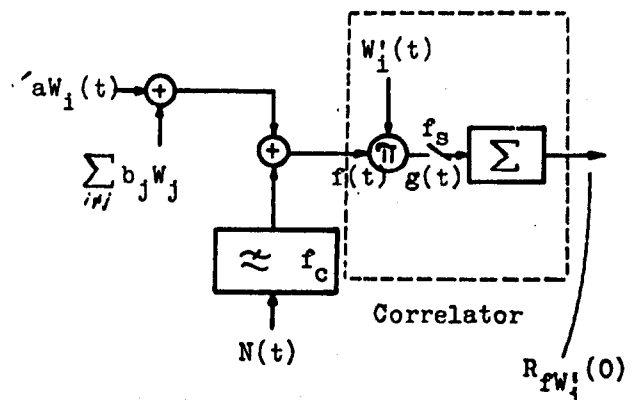


Fig. 6: System for noise-calculations

The output of the correlator is called $R_{fw_1'}(0)$ and is the cross-correlation-function of the incoming, noise-corrupted signal and the locally generated Walsh-function W_1' . Since the corrupting noise is a random variable, the correlator output will itself be a random variable. Its value without input-noise, that is with just the function $a \cdot W_1(t)$ as input, is

$$E [R_{fw_1'}(0)] = a$$

The additive noise gives the random variable a variance σ^2 , which is dependent of f_c (low-pass-filter cut-off frequency), f_s the sampling frequency f_s and the correlation duration.

This variance turns out to be ([4], p 282)

$$\sigma^2 = \sum_{k=-(n-1)}^{n-1} \frac{n-|k|}{n^2} \left[R_{gg}\left(\frac{k}{f_s}\right) - E^2[g(t)] \right]$$

σ^2 is the noise power at the output of the correlator.

Therefore the signal to noise power ratio at the output becomes:

$$\frac{S'}{N'} = \frac{E^2[R_{fW_1}(0)]}{\sigma^2}$$

The signal to noise ratio then can be rewritten:

$$\frac{S'}{N'} = \frac{E^2[R_{fW_1}(0)]}{\sum_{k=-(n-1)}^{n-1} \frac{n-|k|}{n^2} R_{W_1 W_1}\left(\frac{k}{f_s}\right) \cdot R_{NN}\left(\frac{k}{f_s}\right)}$$

From this equation S'/N' can be calculated. The calculation is rendered easier if

$$R_{W_1 W_1}\left(\frac{k}{f_s}\right)$$

is assumed as the constant value

$$R_{W_1 W_1}(0) = a \text{ (worst-case)}$$

Furthermore, $R_{NN}\left(\frac{k}{f_s}\right)$ is given by

$$R_{NN}(t) = 4\pi f_c \rho_{NN} \frac{\sin(2\pi f_c t)}{2\pi f_c}$$

ρ_{NN} being the double-sided power density of the white Gaussian noise.

The calculation of $\frac{S'}{N'}$ was carried out, under the above worst-case condition, for a cut-off frequency of the low-pass-filter chosen at

$$f_c = 1/t_r$$

(t_r = shortest pulse duration of highest Walsh-function).

The calculated values of the signal to noise improvement γ are given in Fig. 7, in function of the number of samples per Walsh-function period where γ is obtained from S'/N' , the equation:

$$\gamma = \frac{S}{N} \frac{N'}{S'} = \frac{a^2}{2\rho_{NN} 2\pi f_c} \cdot \frac{N'}{S'}$$

In the same figure values measured on a test system have been plotted.

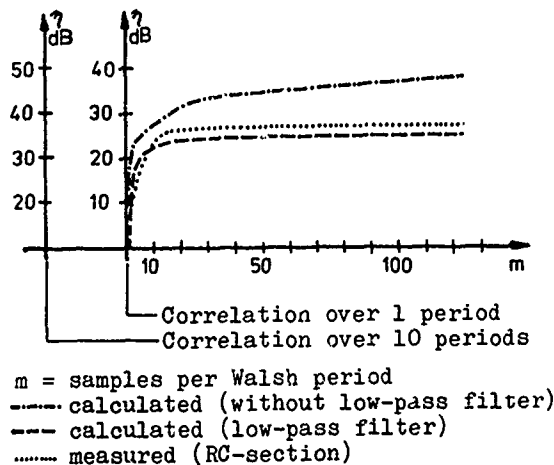


Fig. 7: Signal to noise power ratio improvement

If now several Walsh-functions are present ($s(t) \neq 0$) neither the expected value $E[g(t)]$ nor the variance σ^2 of the correlator output are affected, if an equal number of samples are taken from every time interval t_r . (This follows from the orthogonality of the Walsh-functions and the linearity of the system of Fig. 6). Thus, the curves shown in Fig. 7 are valid independent of the number of Walsh-functions present on the channel.

5. Conclusions

Walsh-functions have shown to be particularly well suited for a situation where a large number of binary coded signals with a low transmission rate have to be multiplexed in a single, noisy channel. A Walsh system becomes particularly advantageous where miniaturization or high reliability is required because it can easily be implemented by fully-digital MSI's.

With a synchronous transmission and correlation detection over many periods, the main problem would seem to be the transmission of the clock and synchronization pulses from the central monitoring station to the various monitoring points. In the situation described in this paper, it may be assumed that the central monitoring station, which is at the same time the power source of the whole system, is capable of sending out very strong pulses that are hardly disturbed. In another situation, however, where the power of the pulses are limited, this additional problem would have

to be carefully studied.

6. Acknowledgments

The authors wish to thank the Cableworks Brugg, Ltd., Brugg, Switzerland, who have supported the work described herein, and especially B. Capol and B. Weber with whom it has been a pleasure to cooperate.

The authors also wish to thank Prof. Dr. E. Baumann, director of the Institute of Applied Physics, under whose guidance this project was carried out, for his encouragement and his farsighted stress on applied research.

7. References

- [1] Capol, B.: "Hochleistungskabel für den Energietransport", Technische Rundschau, Nr. 52, December 10, 1971
- [2] Bazzi, G, Calzolari, P, Palmieri N, Bellucci F,: "220 kV cable connection in Rome", IEEE Summer Power Meeting, Portland, Oregon, July 18-23, 1971, Paper No 71 TF 556-PWR.
- [3] Harmuth, H.: "Die Orthogonalteilung als Verallgemeinerung der Zeit- und Frequenzteilung", Archiv der elektrischen Uebertragung, Band 18, Heft 1, 1964.
- [4] Lee, Y.W.: "Statistical Theory of Communication", John Wiley + Sons, New York, 1960.

SOME CONSIDERATIONS IN SEQUENCY MULTIPLEXING SYSTEMS

Chiang Lin and Someshwar C. Gupta
Information and Control Sciences Center
SMU Institute of Technology
Dallas, Texas 75222

Abstract

This paper describes a Sequency Multiplexing System. First a output of a sequency low pass filter (SLPF) in the matrix form is developed for any input by considering the first m -terms of Walsh series expansion of the input. The SLPF is assumed to be composed of an integrator, delay elements, a sampler and a zero-order hold. The number of delay elements is dependent upon m . Using a block impulse as a testing input, the sequency multiplexing system is constructed using the sequency low pass filters. The output of each channel of the system is expressed in a closed mathematical form. The use of the pre and post amplifier in the multiplexing structure is also discussed. Results of the cross talk between the channels are obtained and these are given in a table form. The noise in the channel is also considered.

Introduction

In recent years Walsh function has become quite popular in communication applications. The advantages of Walsh functions are primarily due to the efficiency of implementation and signal manipulation. Digital networks are more acceptable to the practical applications since the invention of IC circuits. For digital networks, pulse type waveform like Walsh functions are more suitable than sinusoidal waveforms. The use of Walsh functions in multiplexing systems have been discussed in previous papers [1,2,3]. In this paper we select a communication model generally used in a multiplexing system for detailed investigation. The derivation of the equations and results are stated.

Representation of Walsh Functions by Matrices

Walsh functions form a complete set of orthogonal functions [4]. The Walsh function $W(m, \theta)$ may be defined by the difference equation

$$W(2n+p, \theta) = (-1)^{(\lfloor n/2 \rfloor + p)} \{ W[n, 2(\theta + \frac{1}{4})] + (-1)^{(n+p)} W[n, 2(\theta - \frac{1}{4})] \} \quad (1)$$

where $p=0$ or 1 ; $n=0, 1, 2, \dots$; $-1/2 < \theta < 1/2$; $\lfloor n/2 \rfloor$ defines the largest integer less than or equal to $n/2$. Define $W(0, \theta)$ as

$$W(0, \theta) = \begin{cases} 1 & \text{for } -1/2 < \theta < 1/2 \\ 0 & \text{for } 1/2 \leq \theta < 3/2 \end{cases} \quad (2)$$

With the above difference equation and $W(0, \theta)$, $W(m, \theta)$, $m=1, 2, \dots$, can be generated in sequence. For example, $W(1, \theta)$ can be obtained by setting $p=1$ and $n=0$. From eq. (1)

$$W(1, \theta) = - \{ W[0, 2(\theta + \frac{1}{4})] - W[0, 2(\theta - \frac{1}{4})] \}$$

The function $W(0, 2\theta)$ has the same shape as $W(0, \theta)$ except it is squeezed into $-1/4 \leq \theta < 1/4$.

$W[0, 2(\theta + \frac{1}{4})]$ and $W[0, 2(\theta - \frac{1}{4})]$ translate $W(0, 2\theta)$ to the left and right by $1/4$. After $W(1, \theta)$ is found, $W(2, \theta)$ can be generated by letting $n=1$, $p=0$. $W(3, \theta) \dots$ etc.

Fig. (1) shows samples of Walsh function. The independent variable is the normalized time $\theta = \frac{t}{T_0}$. T_0 is the smallest orthogonal period for Walsh functions. Since Walsh functions are pulse-type waveforms with amplitude $+1$ or -1 , it is more convenient to represent the subset $\{W(0, \theta), \dots, W(m, \theta)\}$ within one orthogonal interval T_0 by an $(m+1) \times (2^r)$ matrix, where r can be calculated by the inequality

$$2^{r-1} < (m+1) \leq 2^r \quad (3)$$

For example:

$$\begin{aligned} m=0 \quad W_0 &= \begin{bmatrix} + \end{bmatrix} & m=1 \quad W_1 &= \begin{bmatrix} + & + \\ - & + \end{bmatrix} \\ m=2 \quad W_2 &= \begin{bmatrix} + & + & + & + \\ - & - & + & + \\ - & + & + & - \end{bmatrix} & m=3 \quad W_3 &= \begin{bmatrix} + & + & + & + \\ - & - & + & + \\ - & + & + & - \\ + & - & + & - \end{bmatrix} \end{aligned}$$

where $+$ and $-$ means $+1$ and -1 and each column represents a $T_0/2^r$ interval.

Output of An Ideal Sequency Low Pass Filter (SLPF)

The Walsh series expansion of a function $F(\theta)$, $-1/2 \leq \theta < 1/2$, can be expressed as

$$F(\theta) = \sum_{i=0}^{\infty} a(i) W(i, \theta); \quad a(i) = \int_{-1/2}^{1/2} F(\theta) W(i, \theta) d\theta \quad (4)$$

The Walsh series expansion has the same properties as the Fourier series expansion, except when $F(\theta)$ extends over the range of $-1/2 \leq \theta < 1/2$. then the Walsh series expansion must be recalculated on each orthogonal interval such as $1/2 \leq \theta < 3/2$, $3/2 \leq \theta < 5/2$, etc.

Let $F(\theta)$ pass through a sequency filter, the general expression for the output can be written as

$$F_0(\theta) = \sum_{i=a}^b a(i) K(i) W[i, \theta - \theta(i)] \quad (5)$$

The type (low pass, high pass, bandpass) of sequency filter depends on the values of a and b . $K(i)$ and $\theta(i)$ determine the characters of the sequency filter. $K(i)$ is called the attenuation factor and $\theta(i)$ is called time delay.

The difference between the ideal frequency and sequency filters is stated as follows. In the frequency sense, the ideal filter can be defined by having a flat amplitude and linear phase responses. However, we defined "the sequency as the generalized frequency," [5] hence, the "phase" is not meaningful in a sequency sense. Instead the ideal sequency filter is defined as a filter with a flat amplitude response and a constant time delay.

The output of an ideal sequency low pass filter can be obtained by inserting the following

values in Eq. (5): $a=0$, $b=m$; $K(0)=K(1)=\dots=K(m)=1$;
 $\theta(0)=\theta(1)=\dots=\theta(m)=1$ (6)

That means the sequence low pass filter passes the first $m+1$ terms with a unity attenuation and time delay. Eq. (5) now becomes

$$F_0(\theta) = a(0)W(0, \theta-1) + a(1)W(1, \theta-1) + \dots + a(m)W(m, \theta-1)$$

With the help of Eq. (4), the form of Eq. (7) can be rewritten as

$$F_0(\theta) = \left[\int_{-1/2}^{1/2} F(\theta)W(0, \theta)d\theta \quad \int_{-1/2}^{1/2} F(\theta)W(1, \theta)d\theta \quad \dots \quad \int_{-1/2}^{1/2} F(\theta)W(m, \theta)d\theta \right] \begin{bmatrix} W(0, \theta-1) \\ W(1, \theta-1) \\ \vdots \\ W(m, \theta-1) \end{bmatrix} \quad (8)$$

Each element in the row matrix may be considered as the summation of 2^F subintegrals. In each subinterval $W(m, \theta)$ is a constant with value +1 or -1. The column matrix is a Walsh matrix with a unit time delay. It does not effect the results if we calculate Eq. (8) in current time, then deally the whole waveform by 1.

A more convenient form of Eq. (8) is

$$F_0(\theta) = \left[\int_{-1/2}^{1/2} F(\theta)d\theta \quad \dots \quad \int_{-1/2}^{1/2} F(\theta)d\theta \quad \dots \quad \int_{-1/2}^{1/2} F(\theta)d\theta \right] \begin{bmatrix} W(0, \theta) \\ W(1, \theta) \\ \vdots \\ W(m, \theta) \end{bmatrix} \quad (9)$$

For convenience, let $D=2^F$

$$F_0(\theta) = V W_m^T W_m \Big|_{\theta=0-1} \quad \text{where} \quad V = \left[\int_{-1/2}^{1/2} F(\theta)d\theta \quad \dots \quad \int_{-1/2}^{1/2} F(\theta)d\theta \quad \dots \quad \int_{-1/2}^{1/2} F(\theta)d\theta \right] \quad (10)$$

Eq. (10) shows the most general form of the output of an ideal sequence low pass filter. In a real situation $F(\theta)$ is divided into time section $-1/2 < \theta < 1/2$, $1/2 < \theta < 3/2$, The outputs of the filter are calculated on each time base. Such a division does not place any restriction on the signal $F(\theta)$. However, a synchronization signal is required from which the beginning and end of the intervals can be derived. The results of the first few $W_m^T W_m$ are presented as follows

$$W_0^T W_0 = I; \quad W_1^T W_1 = 2I; \quad W_2^T W_2 = \begin{bmatrix} 3 & 1 & -1 & 1 \\ 1 & 3 & 1 & -1 \\ -1 & 1 & 3 & 1 \\ 1 & -1 & 1 & 3 \end{bmatrix}$$

$$W_3^T W_3 = 4I$$

$$\text{when } m = 2^F - 1 \quad W_m^T W_m = 2^F I$$

Example:

For explanation purpose the input waveform S_1 is picked as step, ramp and parabolic function in each interval. The output waveforms calculated from Eq. (10) with $m=0, 1, 2, 3$ are plotted in Fig. 2.

Circuits for Ideal Sequence Low Pass Filter (SLPF)

The basic lay out of a SLPF according to Eq. (10) is shown in Fig. 3.

The integrator integrates the input function $F(\theta)$ with time interval for integration which depends upon the value of m . In the circuit the time interval for integration is determined by the operating speed of the switches P_1 and P_2 . Practically the operating speeds of P_1 and P_2

are set equal to T_0/D sec. Switch P_2 samples the value of the integrator at the end of each T_0/D sec. P_1 resets the initial condition of the integrator, then the integration for the next T_0/D sec. starts. The switch P_3 sends the sampled values to each delay element. The period for a complete full circle of switch P_3 is T_0 sec. It is obvious that an accurate time reference for switches P_1 , P_2 , and P_3 is required. At the end of each T_0 sec the $D=2^F$ samples values from P_2 are available to the matrix multiplier simultaneously. The output of the matrix multiplier is a row matrix. In the time domain this row matrix is generated in columnwise order. The ZOH holds each value from the matrix multiplier for T_0/D sec. The step function like waveform at the output of the ZOH is $F_0(\theta)$.

The special case for SLPF occurs when $m=2^F-1$. In this case $W_m^T W_m = 2^F I$ always holds. This relation not only simplifies the mathematical manipulation but also simplifies the hardware associated with a practical sequence low pass filter. The modified SLPF for this case is shown in Fig. 4. It is apparent that this is the most desirable type of the SLPF.

Multiplex System by Using Walsh Functions

It is often desirable to transmit several messages on one transmission facility. The process is called multiplexing. Multiplexing system which utilize sinusoidal carrier or subcarriers have been studied extensively in communication theory. The use of Walsh function as carrier or subcarrier in multiplexing systems has recently been studied by Harmuth [4] and Hübner [1-3]. In this section we derive the output for each channel. The related topic such as pre and post amplifier, crosstalk, etc., are discussed in the following sections.

Fig. 5 shows the block diagram of a multiplex system which uses Walsh functions as subcarriers. The set of Walsh subcarriers is written in matrix form.

$$WS = \begin{bmatrix} WS_1 \\ \vdots \\ WS_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1D} \\ a_{21} & a_{22} & \dots & a_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nD} \end{bmatrix} \quad (11)$$

where $D = 2^F$. The input signal S_{T1} is an impulse block with amplitude 1 and duration 1. This signal is the basic "building block" for various Walsh functions. The general performance of the system can be roughly described by using a signal as an input. The signal waveforms at various stages of the system can be easily obtained with the help of Eq. (10) and Fig. 2. S_{T1} passes through a pre-amplifier (PRA) with an adjustable amplification factor A_1 . Therefore, S_{11} has the same shape as S_{T1} except the amplitude times A_1 . For the special input signal S_{T1} , the output of the SLPF has the same waveform as the input. This fact can be easily observed from Fig. 2. Therefore, S_{21} has exactly the same shape as S_{11} except that S_{21} is delayed by T_0 . After multiplying by the Walsh matrix in Eq. (11) S_3 is represented in matrix form as

$$S_3 = \begin{bmatrix} A_1 a_{11} & A_1 a_{12} & \dots & A_1 a_{1D} \\ A_1 a_{21} & A_1 a_{22} & \dots & A_1 a_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ A_n a_{n1} & A_n a_{n2} & \dots & A_n a_{nD} \end{bmatrix} \quad (12)$$

Since each column dominates a T_0/D interval, the signal S_4 can be obtained by row wise summation of S_3

$$S_4 = \left[\sum_{p=1}^n A_p a_{p1} \quad \sum_{p=1}^n A_p a_{p2} \quad \dots \quad \sum_{p=1}^n A_p a_{pD} \right]$$

$$S_4 = \sum_{h=1}^n \sum_{p=1}^n A_p a_{ph} \{u[\theta - (1 + \frac{h-1}{D})] - u[\theta - (1 + \frac{h}{D})]\} \quad (13)$$

The actual waveform of S_4 is shown in Fig. 6.

Theoretically, we can feed the signal S_4 directly into the communication channel. Due to the step function like waveform as shown in Fig. 6, the wide spread of the frequency spectrum of the signal S_4 and the continually increasing demands for frequency allocations as function of n provide undesirable properties for direct transmission. An ideal frequency low pass filter (LPF), with normalized cut-off frequency F_c , has been placed at the final stage of the transmitter for the purposes of reducing the bandwidth requirement for the whole multiplex system and reducing the interference to other communication system.

The output signal, S_5 , at the transmitter may be obtained by taking the Fourier transform of S_4 and then taking the inverse transform with a frequency limitation F_c . The Fourier transform of an individual "step element"

$$\sum_{p=1}^n A_p a_{ph} \{u[\theta - (1 + \frac{h-1}{D})] - u[\theta - (1 + \frac{h}{D})]\}$$

is represented by the amplitude response $A(f)$ and phase response $\phi(f)$.

$$A(f) = \frac{1}{D} \frac{\sin \frac{\pi f}{D}}{\frac{\pi f}{D}} \sum_{p=1}^n A_p a_{ph} ; \phi(f) = e^{-j2\pi f(1 + \frac{2h-1}{2D})} \quad (14)$$

Taking the inverse Fourier transform of S_4 with frequency limitation F_c yields for the output of the LPF

$$S_5 = \sum_{h=1}^n \sum_{p=1}^n A_p a_{ph} \frac{1}{D} \int_{-F_c}^{F_c} \frac{\sin \frac{\pi f}{D}}{\frac{\pi f}{D}} e^{-j2\pi f(1 + \frac{2h-1}{2D})} e^{j2\pi f \theta} df$$

$$= \frac{1}{\pi} \sum_{h=1}^n \sum_{p=1}^n A_p a_{ph} \{S_1[2\pi F_c(\theta - 1 - \frac{h-1}{D})] - S_1[2\pi F_c(\theta - 1 - \frac{h}{D})]\} \quad (15)$$

where $S_j(z) = \int_0^z \frac{\sin t}{t} dt \quad [6] \quad (16)$

Assuming the communication channel is ideal which means the channel introduces constant attenuation and time delay to all frequencies, compensation for constant attenuation can be achieved by either the pre or the post amplifier. Detection of the signal in the presence of variable time delay may be accomplished by sending synchronous signals across the channel. This sets up the exact timing between the transmitter and receiver. Hence, it is reasonable to assume that the signal passes through the channel without any distortion.

Multiplying S_5 by the Walsh matrix WS gives

$$S_6 = \begin{bmatrix} S_5^{a_{11}} & \dots & S_5^{a_{1a}} & \dots & S_5^{a_{1D}} \\ S_5^{a_{j1}} & \dots & S_5^{a_{ja}} & \dots & S_5^{a_{jD}} \\ S_5^{a_{n1}} & \dots & S_5^{a_{na}} & \dots & S_5^{a_{nD}} \end{bmatrix} \quad (17)$$

At the receiver assume that the SLPF with $m=0$, the output of the SLPF, S_7 , can be calculated by applying Eq. (10)

$$S_{7j} = \int_1^2 [S_5^{a_{j1}} \dots S_5^{a_{ja}} \dots S_5^{a_{jD}}] d\theta \Big|_{\theta=0-1}$$

$$= \sum_{a=1}^D a_{ja} \int_{1+\frac{a-1}{D}}^{1+\frac{a}{D}} \frac{1}{1+\frac{a-1}{D}} \sum_{h=1}^n \sum_{p=1}^n A_p a_{ph} [S_1 2\pi F_c(\theta - 1 - \frac{h-1}{D}) - S_1 2\pi F_c(\theta - 1 - \frac{h}{D})] d\theta \Big|_{\theta=0-1} \quad (18)$$

By memorizing the time delay introduced by SLPF, we omit the time delay sign $\theta=0-1$. Closed form integration is obtained by applying the following formula

$$\int S_1(x) dx = x S_1(x) + \cos x + c \quad (19)$$

S_{Rj} equals S_{7j} times B_j . Therefore, the outputs of the entire system are written as

$$S_{Rj} = \frac{B_j}{\pi} \sum_{a=1}^D \sum_{h=1}^n \sum_{p=1}^n a_{ja} A_p a_{ph} [H_1(a, h) + H_2(a, h)]$$

$$H_1(a, h) = \left(\frac{a-h+1}{D}\right) S_1 2\pi F_c \left(\frac{a-h+1}{D}\right) + \left(\frac{a-h-1}{D}\right) S_1 2\pi F_c \left(\frac{a-h-1}{D}\right) - 2 \left(\frac{a-h}{D}\right) S_1 2\pi F_c \left(\frac{a-h}{D}\right)$$

$$H_2(a, h) = \frac{1}{2\pi F_c} [\cos 2\pi F_c \left(\frac{a-h+1}{D}\right) + \cos 2\pi F_c \left(\frac{a-h-1}{D}\right) - 2 \cos 2\pi F_c \left(\frac{a-h}{D}\right)] \quad (20)$$

Fig. 7 shows the output of each channel ($n=4$). S_{jj} is calculated by setting $A_j=1$, $A_p=0$ if $j \neq p$, $B_j=1$ with $j=1, 2, 3, 4$. In Fig. 8 S_{jj}^2 is obtained by letting $A_j=B_j=1$ with $j=1, 2, 3, 4$. Figs. 9 and 10 are similar to Figs. 7 and 8 except that six channels are used.

The Use of Pre and Post Amplifiers

From Fig. 7, 8, 9, and 10 we observed that the performance of each channel is different. The situation becomes worse as the number of signals increases. This is the case as in telephony multiplex systems when combining twelve channels as base-group, three base-group as super-group, etc. The final performance of each individual channel in such systems will be far different. This circumstance should be improved if the values of A_p in the pre-amplifier of the transmitter and B_j in the post-amplifier of the receiver can be adjusted. In order to get the same quality in each channel, we solve the following simultaneous linear equations

$$\pi C = B_j \sum_{a=1}^D \sum_{h=1}^n \sum_{p=1}^n a_{ja} A_p a_{ph} [H_1(a, h) + H_2(a, h)] \quad j=1, 2, \dots, n \quad (21)$$

In Table 1 we show the possible values of the pre-amplifiers and post-amplifiers for four and six signal channels. The assumptions for calculation are: (1) $c=1$. For the case $c \neq 1$ the values of A_p and B_j should change proportionally. This relation can be obviously seen from Eq. (21). (2) When we adjust the pre-amplifiers, all the post-amplifiers have value 1. The relation being reversed when we adjust the post-amplifiers.

The negative values for the pre and post amplifiers can be explained as at the cut-off frequency F_c the crosstalks dominate the output (discussed

in next section). In multiplex systems this is undesirable. Therefore, we choose F_c such that all the values of the pre and post amplifiers are positive.

Discussion of Channel Crosstalks

The crosstalks between channels can be obtained directly from Eq. (20). S_{ij} is defined as the crosstalk between channels i and j . Its value means that with an input at channel i of the transmitter, we calculate the output at channel j of the receiver. The symmetrical relation $S_{ij}=S_{ji}$ become obvious after we discuss Eq. (20) in detail.

From Eq. (20) we write out S_{ij} directly

$$\pi S_{ij} = \sum_{a=1}^D \sum_{h=1}^D a_{ja} a_{ih} [H_1(a, h) + H_2(a, h)] \quad (22)$$

A better way to manipulate this summation is to write out the following matrix

$$\begin{bmatrix} a_{11} \\ a_{12} \\ \vdots \\ a_{1D} \end{bmatrix} \begin{bmatrix} a_{j1} \\ a_{j2} \\ \vdots \\ a_{jD} \end{bmatrix}^T = \begin{bmatrix} a_{j1}a_{11} & a_{j1}a_{12} & \dots & a_{j1}a_{1D} \\ a_{j2}a_{11} & a_{j2}a_{12} & \dots & a_{j2}a_{1D} \\ \vdots & \vdots & \ddots & \vdots \\ a_{jD}a_{11} & a_{jD}a_{12} & \dots & a_{jD}a_{1D} \end{bmatrix} \quad (23)$$

This matrix is the multiplication of two rows in the Walsh matrix with all the entries +1 or -1. The summation of Eq. (22) equals the summation of each individual element in matrix times its respected H 's functions.

Observe from Eq. (20) that $H_1(a, h)$ and $H_2(a, h)$ are symmetrical with respect to a and r . In addition, they are even functions of $(a-r)$. The interchange of i and j in Eq. (23) does not change the result of the summation. By use of the aforementioned properties, it can be shown that $S_{ji} = S_{ij}$.

By defining $d = |a-h|$

$$H(d) = H_1(a, h) + H_2(a, h) = H_1(|a-h|) + H_2(|a-h|) \quad (24)$$

then the summation of Eq. (22) becomes a summation of each element in the following matrix

$$\begin{bmatrix} a_{j1}a_{11}H(0) & a_{j1}a_{12}H(1) & \dots & a_{j1}a_{1D}H(D-1) \\ a_{j2}a_{11}H(1) & a_{j2}a_{12}H(0) & \dots & a_{j2}a_{1D}H(D-2) \\ \vdots & \vdots & \ddots & \vdots \\ a_{jD}a_{11}H(D-1) & a_{jD}a_{12}H(D-2) & \dots & a_{jD}a_{1D}H(0) \end{bmatrix} \quad (25)$$

From Eq. (25) we get the "free of crosstalk" condition

$$\sum_{ja} a_{ih} = 0 \quad \text{with } |a-h|=d; d=0, 1, 2, \dots, D-1 \quad (26)$$

Example: Calculate the crosstalk between channels when $i=4, j=4$.

From the Walsh matrix shown in the first section, we calculate the crosstalk between channels 1 and 2 by using Eq. (23).

$$\begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \end{bmatrix} \begin{bmatrix} a_{21} \\ a_{22} \\ a_{23} \\ a_{24} \end{bmatrix}^T = \begin{bmatrix} + \\ + \\ + \\ + \end{bmatrix} [- - + +] \rightarrow \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Using the criterion of Eq. (26) we show $S_{12}=0$.

By the same token the crosstalk between channels 1 and 3 can be calculated by

$$\begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \end{bmatrix} \begin{bmatrix} a_{31} \\ a_{32} \\ a_{33} \\ a_{34} \end{bmatrix}^T = \begin{bmatrix} + \\ + \\ + \\ + \end{bmatrix} [- + + -] \rightarrow \begin{bmatrix} 2H(1) \\ -2H(3) \\ 0 \\ 0 \end{bmatrix}$$

Therefore, $S_{13} = 2H(1) - 2H(3)$. The crosstalks between other channels are shown in Table 2.

Included in the Table are crosstalks when $i=8$ and $j=8$. Fig. 11 shows the crosstalks between channels with values of i and j up to 6.

Noise Consideration in Multiplexing System

In the previous sections the multiplex system without medium noise was considered. When the communication channel noise is introduced some of the equations require changes. A review of the system in Fig. 5 shows that the medium noise is added to S_5 . The equations requiring modification are

$$S_5' = \frac{1}{\pi} \sum_{h=1}^D \left[\sum_{p=1}^n A_p a_{ph} \{ S_1 [2\pi F_c (\theta - 1 - \frac{h-1}{D})] - S_1 [2\pi F_c (\theta - 1 - \frac{h}{D})] \} + n_h \right] \quad (15')$$

$$S_{7j}' = \sum_{a=1}^D a_{ja} \int_{1+\frac{a-1}{D}}^{1+\frac{a}{D}} \frac{1}{1+\frac{a-1}{D}} \sum_{h=1}^D \left[\sum_{p=1}^n A_p a_{ph} \{ S_1 [2\pi F_c (\theta - 1 - \frac{h-1}{D})] - S_1 [2\pi F_c (\theta - 1 - \frac{h}{D})] \} + n_h \right] d\theta \quad (18')$$

$$S_{Rj}' = E\{B_j \cdot S_{7j}'\} \quad (20')$$

where $E\{x\}$ means the expected value of x . The sample noise waveform in the orthogonal interval is presented as

$$n = \sum_{h=1}^D n_h$$

Assuming that n is stationary with $E\{n\} = N$, Eqs. (18) through (20') yield

$$S_{Rj}' = S_{Rj} + \frac{B_j}{\pi} \sum_{a=1}^D a_{ja} \int_{1+\frac{a-1}{D}}^{1+\frac{a}{D}} E\{n\} d\theta$$

$$S_{Rj}' = \begin{cases} S_{Rj} + \frac{B_j}{\pi} N & j = 1 \\ S_{Rj} & j \neq 1 \end{cases} \quad (27)$$

In the above equation we applied the basic property of Walsh functions, i.e., in the Walsh matrix the summation of each row equal to zero except $W(0,0)$. We saw from Eq. (27) that the medium noise introduces distortion only limited to the first message channel which use $W(0,0)$ as the carrier. This result is caused by the combined use of the Walsh function and SLPF. This result is not obtained in frequency multiplexing systems, because the LPF in each message channel passes a portion of the noise whose frequencies are located within the signal frequency band. This is one of the advantages in using frequency multiplexing systems.

REFERENCES:

1. H. Hüber, "Analog and Digital Multiplexing by Means of Walsh Functions," *Applications of Walsh Function Symposium*, 1970, pp. 236-247.
2. H. Hüber, "Multiplex System Using Sums of Walsh Functions as Carriers," *Applications of Walsh Functions Symposium*, 1971, pp. 180-191.
3. H. Hüber, "On The Transmission of Walsh Modulated Multiplex Signals," *Applications of Walsh Functions Symposium*, 1970, pp. 41-45.
4. H. F. Harmuth, *Transmission of Information by Orthogonal Functions*, Springer-Verlag, Berlin/New York, 1969.
5. H. F. Harmuth, "A Generalized Concept of Frequency and Some Applications," *IEEE Transactions on Information Theory*, Vol. IT-14, pp. 375-382, 1968.
6. E. Jahnke and F. Ende, *Table of Function*, Dover, New York, 1945.

TABLE 1
Various Values of Pre and Post Amplifiers

4 Channels						
F_c	1	2	3	4	5	6
A_1	0.289	0.972	1.01	0.999	0.991	0.998
A_2	36.0	1.30	1.14	1.11	1.10	1.07
A_3	3.76	1.28	1.14	1.11	1.10	1.07
A_4	119.0	1.95	1.23	1.23	1.23	1.16
B_1	0.909	.990	1.02	1.00	0.994	0.999
B_2	2.23	1.26	1.14	1.11	1.10	1.07
B_3	2.23	1.26	1.14	1.11	1.10	1.07
B_4	-7.95	2.03	1.23	1.23	1.24	1.16
6 Channels						
A_1	0.708	1.26	2.05	1.11	1.03	1.02
A_2	0.089	32.7	1.20	1.67	1.20	1.13
A_3	0.907	1.04	1.13	0.999	1.06	1.06
A_4	-347	3.55	1.30	1.43	1.21	1.17
A_5	-312	1.39	1.43	1.42	1.23	1.16
A_6	-502	73.3	1.36	2.76	1.46	1.28
B_1	0.938	1.12	1.07	1.03	1.01	1.01
B_2	2.98	1.91	1.38	1.23	1.17	1.11
B_3	2.37	1.02	1.12	1.08	1.06	1.07
B_4	-9.93	3.30	1.25	1.28	1.29	1.16
B_5	-18.0	2.62	1.43	1.30	1.24	1.21
B_6	-16.7	-3.67	2.61	1.72	1.49	1.20

TABLE 2
The Crosslinks Between Channels with the Values of i and j up to 6

s_{12}	s_{13}	s_{14}	s_{15}	s_{16}	s_{17}	s_{18}
0	$28(1)+28(3)$	0	$-28(1)+28(3)$	0	$28(2)+28(3)$	0
	s_{23}	s_{24}	s_{25}	s_{26}	s_{27}	s_{28}
	0	$-28(1)+28(3)$	0	$-48(1)+48(3)$	0	$-28(1)+28(3)$
		s_{34}	s_{35}	s_{36}	s_{37}	s_{38}
		0	$28(1)+48(3)$	0	$0*(1)+48(3)$	0
			s_{45}	s_{46}	s_{47}	s_{48}
			0	$-28(1)+48(3)$	0	$28(1)+28(3)$
				s_{56}	s_{57}	s_{58}
				0	$28(1)+48(3)+28(3)$	0
					s_{67}	s_{68}
					0	$-48(1)+48(3)+28(3)$
						s_{78}
						0

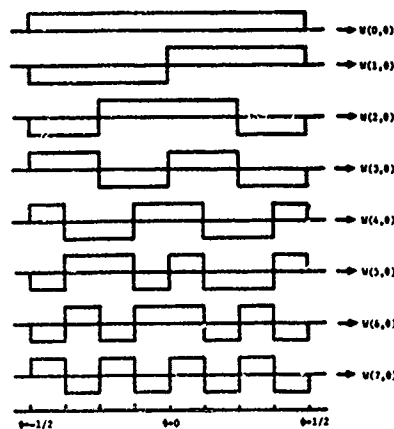


Figure 1: A subset of Walsh function with independent variable $8n/7$. The amplitudes are either +1 or -1.

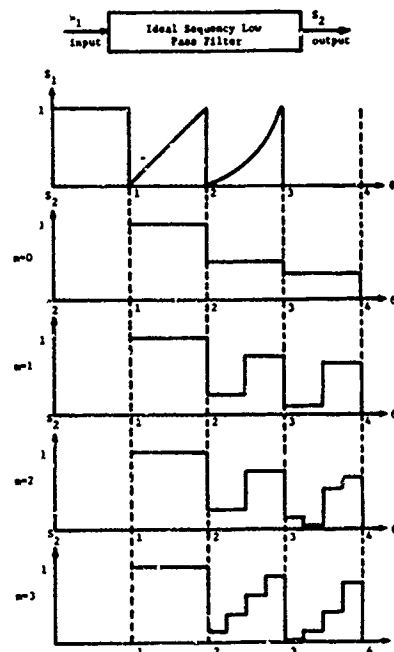


Figure 2: The outputs of the ideal sequence low pass filter (SLPF) with $m=0,1,2,3$.

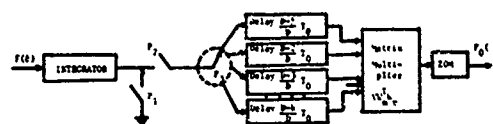


Figure 3: Ideal Sequence Low Pass Filter (SLPF) Invert
Operating Speed: $F_1 = F_2 = F_0/D$
 $F_3 = F_0$
ZOH: Zero-Order-Hold Circuit

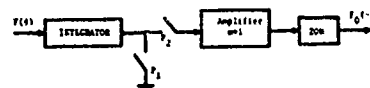


Figure 4: Special Case for Ideal Sequence Low Pass Filter (SLPF)
 $m = 2^k - 1 = 1$
Operating Speed: $F_1 = F_2 = F_0/D$
ZOH: Zero-Order-Hold Circuit

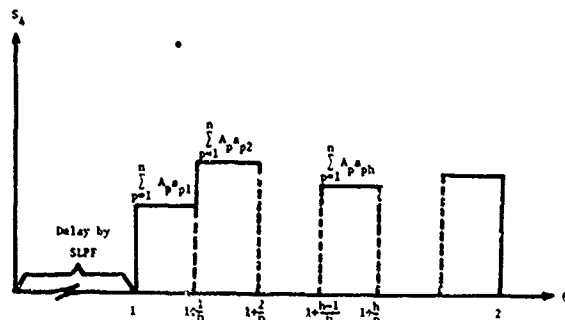
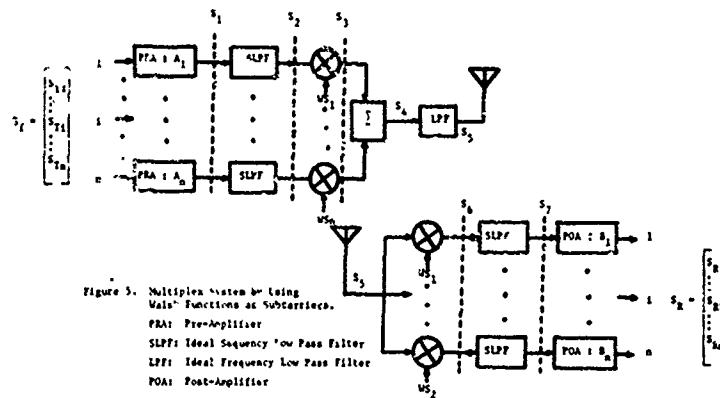


Figure 6: The Signal Waveform of S_k

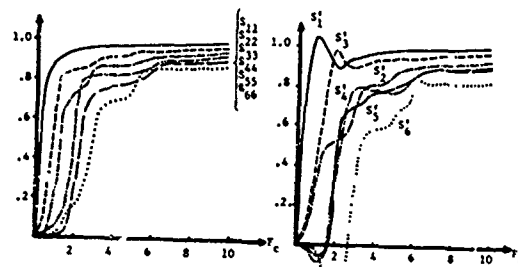


Figure 9: S_{jj} is calculated by $A_j=1, A_p=0$ if $j \neq p$; $B_j=1$ with $j=1,2,3,4,5,6$

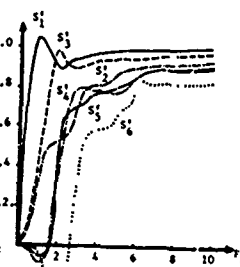


Figure 10: S_j^i is calculated by $A_j=B_j=1, j=1,2,3,4$

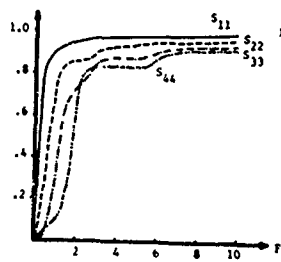


Figure 7: S_{jj} is calculated by $A_j=1, A_p=0$ if $j \neq p$; $B_j=1$ with $j=1,2,3,4$.

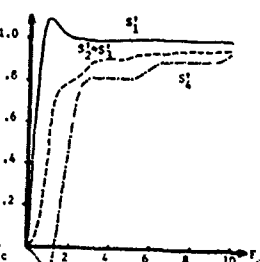


Figure 8: S_j^i is calculated by $A_j=B_j=1, j=1,2,3,4$.

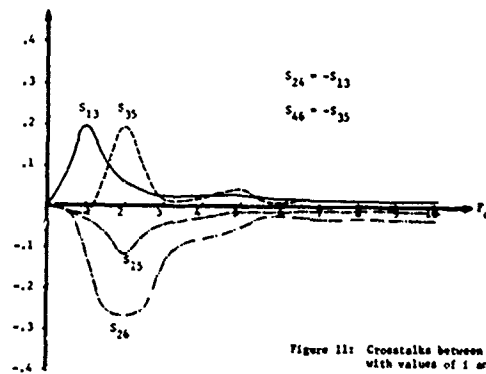


Figure 11: Crosstalks between Channels with values of i and j up to 6.

COMPARISON OF METHODS FOR MULTIPLEXING DIGITAL SIGNALS USING SEQUENCY TECHNIQUES

H. Hübner
Research Institute
of the
Telecommunication Engineering Centre
of the Deutsche Bundespost
Darmstadt, Federal Republic of Germany

Introduction

In communication engineering the methods applied for multiplexing analogue signals using Walsh functions as carriers have proved useful in principle. In their performance they are comparable to carrier frequency systems [1, 2]. They also allow the multiplexing of digital signals. For this type of signal, however, where the information is available in a quantized form more specific methods are known. All of them can be derived from a common system concept as shown in Fig. 1.

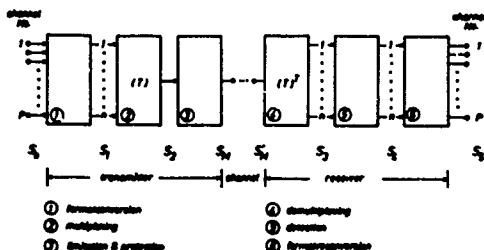


Fig. 1: General concept of transmission system

According to it, source-coded digital-signals of p channels with i binary elements each reach, within a time interval T_0 , a stage which, by format conversion, takes the signal vector S_0 into the system input vector S_1 with n rows and j columns. This conversion is only necessary if the format of S_0 does not correspond with that of S_1 for which

the multiplex arrangement shows optimum system characteristics. The multiplexing which is performed by combining digital signals with the carrier set T can, in general, be described by the matrix equation

$$(S_2) = (T) \cdot (S_1) \quad (1)$$

In order to adapt the resulting multiplex signal S_2 to the characteristics of the transmission channel it can, in addition, be subjected to a limiting or signal protection process before the elements of the transmit-signal column vector S_M resulting therefrom are, one after the other, transmitted to the receiver input. From the signal S'_M the receiver recovers the channel signals S_3 by an operation which is inverse to the multiplexing. For this purpose the matrix equation

$$(S_3) = (T)^T \cdot (S'_M) \quad (2)$$

is to be performed, it requires the transpose of the carrier matrix to be known. Subsequently, the receiver produces the system output signal S_4 by weighting of S_3 in accordance with system-inherent conditions in the stage for signal detection. After signal format re-conversion S_4 becomes S_5 and

contains the digital signals of the p input channels to be transmitted. With error-free transmission channel there holds

$$S_M' = S_M. \quad (3)$$

Irrespective of the above the entire transmission process will be free of errors if

$$S_4 = S_1 \quad \text{and} \quad S_5 = S_0 \quad (4)$$

The described general transmission concept does not only comprise multiplexing by means of sequency division but, as a special case, also the signal combination and transmission by pure time division and thus shows the close relation between these different multiplex principles. The assignment of j -digit, binary code signals of n channels to n time slots in the multiplex signal, which is common practice in the case of PCM systems, and their re-assignment can be obtained from equations (1) and (2) with $S_2 = S_M'$ if (T) is the identity matrix. Then (T) represents the system of the orthogonal block pulses which serve as carriers in the case of multiplexing with pure time division.

Methods of multiplexing

Multiplexing by carrier modulation

This method is rigorously based on the sequency multiplex principle for analogue signals [3, 4]. It allows binary signals to be transmitted on Walsh carriers, which can be represented in the time interval T_0 by a one-digit column vector S_1 of the length 2^k with the elements ± 1 . The multiplexing process according to equation (1) is described by the modulation equation

$$(S_2) = (W)(S_1) \quad (5)$$

where (w) is the orthogonal system of 2^k Walsh functions. Multi-digit column vectors S_1 are also permissible. In this case, however, only certain Walsh functions may be used as carriers [5]. If S_2 is not limited, the multiplex signal S_M is obtained. In the case of n channels with $n \leq 2^k$ its amplitude has $(n + 1)$ possible levels. Here, the frequency of occurrence of the individual amplitude levels shows a binomial distribution and is symmetrical to zero. The multiplex signal can be transmitted over any distance on channels with regenerative repeaters without impairment of its information content by cumulative noise. From S_M' one obtains S_3 according to equation (2) by performing the matrix operation

$$(S_3) = 2^{-k}(W)^T \cdot (S_M') \quad (6)$$

After passing through the signal detection stage which normally contains zero-symmetrical limiters, there hold:

$$(S_4) = \text{sign} [(S_3)] \quad (7)$$

Interference signals N which, during transmission, are superimposed on the multiplex signal and are quantized by the regenerative repeaters will remain ineffective as long as the interference D - which is calculated from

$$\begin{aligned} (S_3) &= 2^{-k} \cdot (W)^T \cdot [(S_M') + N] = \\ &= (S_1) + 2^{-k}(W)^T \cdot N = (S_1) + D \end{aligned} \quad (8)$$

according to equation (6) - is suppressed during signal detection in accordance with equation (7).

With $n = 2^k$ channels there are $(n + 1)^n$ different signal patterns in the multiplex signal, only 2^n of which occur in their representation by the input signal vector S_1 . Because of its

multi-level amplitude, S_2 contains a high redundancy component. It is not required for signal recovery in the receiver and can be partly removed by limiters in the signal limiting stage. Here, the number of amplitude levels required for an error-free transmission depends on the number of operated channels and the multiplex signal-to-noise ratio. For a 4- and 8-channel system with the limiting degree α as parameter Fig. 2 shows the immunity of the multiplex equipment against external noise as a function of the number n of active channels. Here, S_t is the minimum threshold distance occurring in the case of all possible signal vectors S_1 to S_4 .

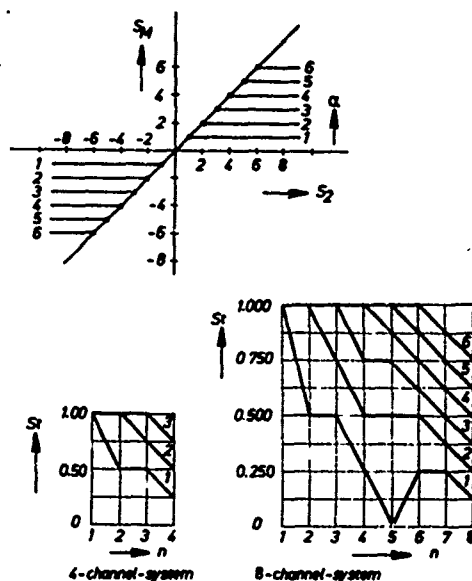


Fig. 2: System immunity

For an error-free transmission the multiplexing by carrier modulation in general requires a higher bit rate than a pure time division multiplex method and a more complex regenerative repeater because the multiplex signal has more than two levels.

Multiplexing by binary addition

The disadvantage of a multi-level multiplex signal is avoided by a computer-aided multiplex method which is based on the multiplexing by carrier modulation but performs the calculating operations required for multiplexing and signal recovery in the binary number system [6, 7]. It is not necessarily confined to Walsh functions as carriers. Any orthogonal carrier sets (T) are permissible if there is the transpose of (T). Because of its two-valuedness, signal S_2 which is produced by multiplexing is well suited for transmission.

If j -digit binary signals per channel form the signal input vector S_1 , the multiplexing process according to equation (5) with 2^k channels results in $(k + j) \cdot 2^k$ binary digits for S_2 . The multiplex signal has a high redundancy component because for a redundancy-free transmission with multiplexing by time division only $j \cdot 2^k$ digits are required. The inevitable redundancy R rises exponentially with increasing number of channels

$$R = k \cdot 2^k \quad (9)$$

The ratio of the number of binary digits produced by the binary addition to the number of digits required for the time division multiplex techniques is described by

$$V_k = 1 + \frac{k}{j} \quad (10)$$

It moves towards unity to such an extent as the number of channels decreases with respect to the length of the binary signals to be transmitted. Equations (9) and (10) illustrate the high redundancy content which results in an uneconomical system design.

When considering the signal characteristics of S_2 it becomes evident that by utilization of signal symmetries the inevitable redundancy can only unessentially be reduced to

$$R' = (k - 1) \cdot 2^k \quad (11)$$

The division by 2^k to be carried out according to equation (6) does not lead to any reduction in digits in the multiplex signal.

Investigations into the system behaviour in the case of interference in the multiplex signal have revealed that its effect depends on the error location in S_M and influences all channels. Here, it becomes apparent that the possible signal patterns of S_M have unequal Hamming distances between each other. Very often they are only separated from each other by the minimum distance 1. The redundancy in the multiplex signal thus allows error detection and correction only in some specific cases.

There is no basic change in the system behaviour if not all of 2^k possible channels are active. The multiplex signal then has $(j + x) \cdot 2^k$ binary digits, where x is a discontinuous quantity which depends on the activity factor. It assumes the value 1, even with mean system loading. On the other hand, there is a decrease in the number of signal patterns of S_M having the Hamming distance 1. In general, there are no new symmetries in the multiplex signal, which can be utilized for bit rate reduction.

Only in the case of 8-channel systems with activity factors $\leq 25\%$ symmetry characteristics develop in the multiplex signal, the utilization of which leads to minimum bit rates. With two active channels they are, however, still 1 bit

per channel above the bit rate required for the method applying pure time division multiplexing.

It becomes obvious that the multiplexing with binary addition produces a two-valued multiplex signal with redundancy which can neither be eliminated nor be sufficiently used for error detection and correction. The multiplex method described above results in a procedure with non-optimum coding.

Multiplexing by majority decision

This principle is a modification of the multiplexing by carrier modulation. It differs from the latter in specially chosen carrier sets and enables digital signals to be transmitted free from redundancy on a limited number of channels or under certain conditions the utilization of system-inherent redundancy for error detection and correction purposes [8, 9].

The input signals of n channels which are combined to form one-digit column vectors S_1 change according to equation (1) the amplitude of the row vectors of (T) serving as carriers. After summation one obtains a signal S_2 whose amplitude has $(n + 1)$ levels. By a clipping symmetrical to zero, it is transferred into the two-valued redundancy-free signal S_M

$$(S_M) = \text{sign}[(S_2)] \quad (12)$$

and transmitted to the receiver input. Demodulation is carried out in the manner normally used in the sequency technique, i.e. by correlation with subsequent threshold weighting of the detected signal. With equation (2) one obtains

$$(S_M) = \text{sign}[(S_3)] \quad (13)$$

The signal transmission is free from errors if number of channels and carrier set meet certain conditions, i.e. the number of channels and the elements of the row vector of the carrier matrix must be odd. Then, the carriers form only a quasi-orthogonal system since the cross-correlation between the individual functions no longer is zero. The information loss increasing with growing number of channels due to amplitude limitation in the multiplex signal and the errors due to disturbed correlation confine the unambiguous signal transmission to a maximum of seven channels. Here, the carriers used either can be derived from reduced Hadamard matrices, which are obtained by removing one of the rows and one of the columns, or they may consist of cyclically shifted 3- and 7-digit binary pseudo-random sequences. Other carrier sets give rise to system errors.

The effect of bit errors in the multiplex signal occurring during transmission varies with the degree of system loading. It has become apparent that the error distribution and correction properties are independent of the carrier system used. Investigations into the behaviour of a fully loaded 7-channel system show that the error effect depends on the values of the disturbed amplitude level in the multiplex signal prior to clipping. If one element in S_M is disturbed, a maximum of 4 channels are falsified. Fig. 3 illustrates the probability $p(i)$ with which, in the case of one error in the multiplex signal, i channels become faulty when all of the 128 possible signal vectors S_1 are equally distributed.

Contrary to the multiplex signal in a fully loaded system, S_M contains redundancy components in a partly loaded system. Under certain conditions

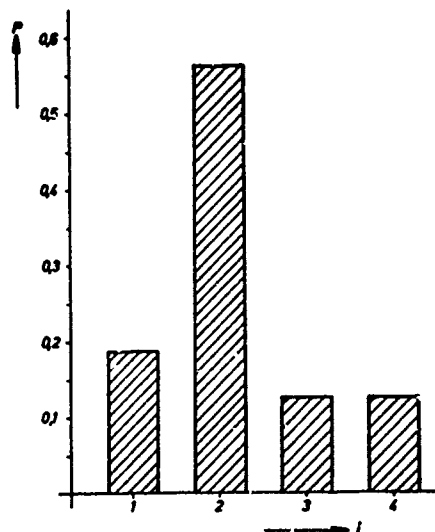


Fig. 3: System error behaviour

this redundancy has an automatic error correction effect. However, error correction only takes place in systems where 4 or 6 channels are disconnected. Otherwise, errors occur in spite of redundancy. A 7-channel system with majority decision corrects up to 3 errors in the multiplex signal if one channel only is active. With 3 channels in operation, one error remains ineffective. If signals from more than 7 channels are to be transmitted by multiplexing with majority decision, groups derived from 3- and 7-channel systems must be used in hierarchical structure. Here, it is advisable to fully load the systems of the lower groups and to assign free channels to the uppermost group if there is the risk of a disturbed transmission. This avoids the possibility of magnified error propagation in the case of fully loaded and cascaded systems.

The multiplexing by majority decision turns out to be a method for the combination of a limited, odd number of channels which enables the signals of a fully loaded system to be transmitted at the same bit rate as in the case of

time division multiplexing. Its error behaviour is poor if the system is fully loaded. Only with one and three active channels it makes optimum use of the redundancy existing in the multiplex signal without requiring additional circuitry for error correction.

Multiplexing by dyadic combination

The calculating operations, addition and subtraction, which are required for multiplexing and which, in the case of the two previously described multiplex methods, were performed in the decimal and binary number systems are here replaced by dyadic combinations [10, 11]. In this method the modulo-2-addition corresponds to the addition and the "Searle" operation to the subtraction. The latter is so defined that for two j-digit binary numbers A and B each the expressions

$$\tilde{A} = A \oplus B; \quad \tilde{B} = A \ominus B \quad (14)$$

allow unequivocal conclusions to be drawn with respect to A and B. This multiplex method permits the combination of 2^k channels. The multiplexing process produces a two-valued multiplex signal without additional digits.

If n_i is the j-digit input signal at channel i, the multiplexing process according to equation (1) for 4 channels yields

$$S_2 = \begin{bmatrix} (n_1 \oplus n_2) \oplus (n_3 \oplus n_4) \\ (n_1 \oplus n_2) \ominus (n_3 \oplus n_4) \\ (n_1 \ominus n_2) \oplus (n_3 \ominus n_4) \\ (n_1 \ominus n_2) \ominus (n_3 \ominus n_4) \end{bmatrix} \quad (15)$$

The vector S_2 can be calculated step by step so that each of the expressions used need be determined only once.

In the receiver the multiplex signal can no longer be separated by means of correlation methods. The multiplexing process represents an unambiguous and thus reversible mapping. Recovery of the signals is, therefore, effected by operations being inverse to \oplus and \ominus .

The dyadic combination may refer to either one or several binary digits. With the one-digit combination, 2 j-digit binary numbers each are combined digit by digit.

$$\begin{aligned} \tilde{A} &= A \oplus B & A &= (a_j, a_{j-1}, \dots, a_1) \\ \tilde{B} &= A \ominus B & B &= (b_j, b_{j-1}, \dots, b_1) \end{aligned}$$

$$\begin{array}{c|c|c} \oplus & b_j & 1 \\ \hline \tilde{a}_j = a_j \oplus b_j & 0 & 1 \\ a_j & 0 & 1 \\ \hline & 1 & 0 \end{array} \quad \begin{array}{c|c|c} \ominus & b_j & 1 \\ \hline \tilde{b}_j = a_j \ominus b_j & 0 & 1 \\ a_j & 0 & 1 \\ \hline & 1 & 0 \end{array} \quad (16)$$

This is a possibility of defining the non-commutative "Searle" operation in such a manner that the binary characters \tilde{A} and \tilde{B} allow conclusions to be drawn with respect to A and B. The separation is effected by the inverse operations:

$$\begin{array}{c|c|c} \text{Inverse } \oplus & \tilde{b}_j & 1 \\ \hline a_j & 0 & 1 \\ \tilde{a}_j & 0 & 1 \\ \hline & 1 & 0 \end{array} \quad \begin{array}{c|c|c} \text{Inverse } \ominus & \tilde{b}_j & 1 \\ \hline b_j & 0 & 1 \\ \tilde{a}_j & 0 & 1 \\ \hline & 1 & 0 \end{array} \quad (17)$$

Even-number input signals can be combined with the two-digit dyadic combination which is performed in pairs:

$$\begin{array}{c|c|c} \oplus & & \\ \hline \tilde{a}_{2i}, \tilde{a}_{2i-1} & b_{2i}, b_{2i-1} & \\ \hline & 00 & 01 \ 11 \ 10 \\ 00 & 00 & 01 \ 11 \ 10 \\ 01 & 01 & 00 \ 10 \ 11 \\ 11 & 11 & 10 \ 00 \ 01 \\ 10 & 10 & 11 \ 01 \ 00 \end{array} \quad \begin{array}{c|c|c} \ominus & & \\ \hline \tilde{b}_{2i}, \tilde{b}_{2i-1} & b_{2i}, b_{2i-1} & \\ \hline & 00 & 00 \ 10 \ 01 \ 11 \\ 00 & 01 & 11 \ 01 \ 10 \ 00 \\ 01 & 11 & 10 \ 00 \ 11 \ 01 \\ 11 & 10 & 01 \ 11 \ 00 \ 10 \end{array} \quad (18)$$

With the aid of the one-digit multiplex law column vectors of the input signal matrix S_1 are mapped to column vectors of the multiplex matrix S_2 . On the other hand, the two-digit method allocates column pairs to each other. This is a coding process. The inverse operations required for decoding two-digit combined signals are defined as well.

If during transmission errors arise in the multiplex signal due to interference signals, the analysis of the system behaviour shows that the interference effect is largely dependent on the error location in S_M . The unambiguous mapping of S_1 to S_M indicates that there is a regularity.

For a fully loaded 4-channel system with multiplexing by one-digit dyadic combination, the upper diagram of Fig. 4 shows the number F_e of disturbed output channels with one falsified multiplex bit as a function of error location. The lower diagram represents the maximum number $F_{e\max}$ of disturbed channels if at the same time e errors occur in S_M . The system behaviour of a fully loaded 8-channel system is shown in Fig. 5 in the same manner.

The channels disconnected by being loaded with a constant zero signal produce redundancy in the multiplex signal which leads to an increased Hamming distance between the remaining multiplex signal patterns. It depends essentially on the position of the inactive channels in the system and, in general, is smaller than in the case of optimum coding. Only one special multiplexing mode in each case allows the maximum possible redundancy utilization. An analysis shows that in the 4-channel system optimum conditions exist with disconnected channels 1 to 3 and that in the 8-channel system the

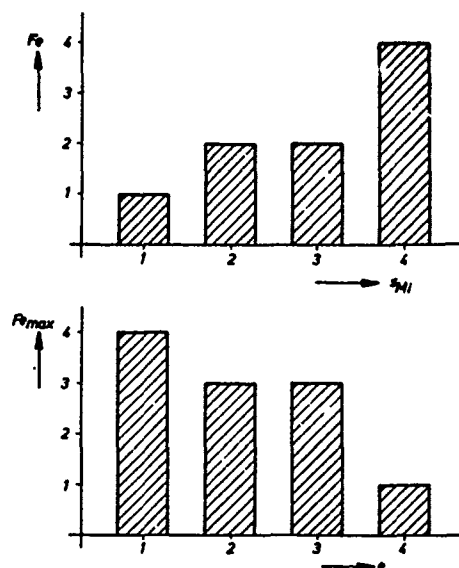


Fig. 4: Error behaviour of the 4-channel system with one-digit dyadic combination

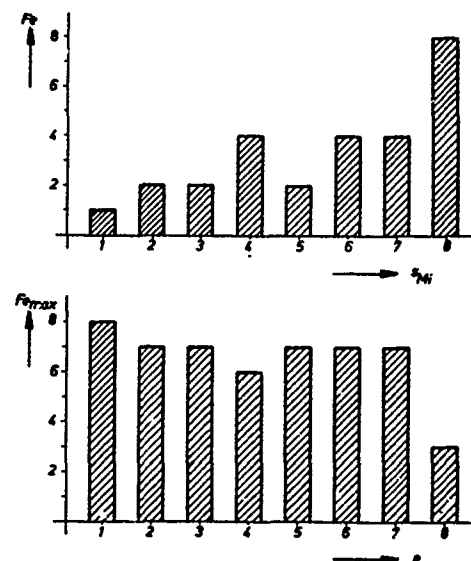


Fig. 5: Error behaviour of the 8-channel system with one-digit dyadic combination

Hamming distance is 8 if only the first channel is active. Here, three errors can be corrected and quadruple disturbances can be detected as it is the case with optimum codes.

For a fully loaded 4-channel system where the multiplexing process is performed by a 2-digit dyadic combination of the input signals, the upper diagram plotted in Fig. 6 shows the number F_e of disturbed receiving channels which occur if the sign of one digit in the multiplex signal is inversed. The abscissa is the error location $S_{M i,j}$. The lower diagram shows the maximum number $F_{e\max}$ of disturbed output channels with e simultaneously occurring errors in the multiplex signal. Fig. 7 illustrates the corresponding behaviour of a fully loaded 8-channel system.

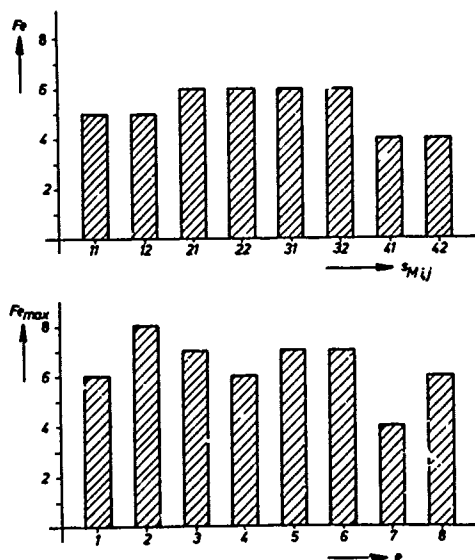


Fig. 6: Error behaviour of the 4-channel system with two-digit dyadic combination

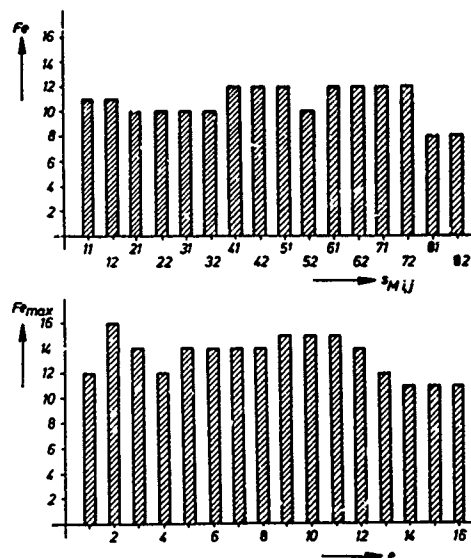


Fig. 7: Error behaviour of the 8-channel system with two-digit dyadic combination

The investigation of the usability of the redundancy that inactive channels produce in multiplex signals formed by 2-digit dyadic combination of the input signals yields optimum conditions for the 4-channel system only with an active fourth channel. The other operating conditions of the 4-channel system as well as all loading possibilities of the 8-channel system do not meet the standard set by the optimum codes.

The multiplexing by dyadic combination proves to be a method which enables digital signals to be combined to a two-valued multiplex signal. The multiplexing and signal separation processes can be described by matrix operations. These are based on formation laws which are equivalent to the coding instructions for linear codes. The error behaviour of fully loaded systems is inferior in comparison to the transmission with pure time division multiplexing. The same applies also to transmission systems having inactive channels. A few systems

with special loading conditions are the only exception. They enable signal protection which is only possible in the case of optimum coding.

Comparison of systems

The described methods allow the transmission of digital signals over a common channel by means of multiplexing. They differ in the required bit rate of the multiplex signal as well as in their error behaviour in the case of transmission disturbances and do not utilize the different redundancy components inherent in the respective system to the same degree for signal protection when the system is fully or only partly loaded.

When using the characteristics of a multiplex equipment with pure time division as a criterion for the suitability of digital communication systems, the multiplexing by carrier modulation does not prove equivalent since it produces a multi-level multiplex signal with an increased bit rate the error-free transmission of which requires expensive regenerative repeaters. This rating will certainly not be changed by the fact that with identical signal-to-noise ratios the two multiplex methods are equally sensitive to white noise, whereas the sequency multiplex system is more insensitive to short pulses as source of interference [5]. Moreover, because of the higher bit rate to be transmitted and the poor utilization of redundancy, the multiplexing by binary addition in general proves inferior to the time division multiplex principle. The multiplexing by majority decision, too, is less advantageous due to low flexibility and poor error behaviour in the case of a fully loaded system. However, it compensates for its disadvantages occurring in connection with

certain minor activity factors by automatic signal protection which requires additional circuitry in the case of time division multiplex techniques. The multiplexing by dyadic combination which, owing to the dyadic operations used is to be included in the sequency technique, is, in general, also disadvantageous because of its poor error behaviour. Here, the characteristics of a time division multiplex system are obtained only under a few, very special operating conditions.

All in all it becomes apparent that the multiplexing of digital signals in the sequency technique with characteristics of normal time division multiplex systems can be performed only under very specific and very limiting conditions.

References

- [1] Proceedings of the 1970 Symposium on Applications of Walsh Functions. April 1970, Washington, USA
Nat. Techn. Inf. Service AD 707 431
- [2] Proceedings of the 1971 Symposium on Applications of Walsh Functions. April 1971, Washington, USA
Nat. Techn. Inf. Service AD 727 000
IEEE Trans. EMC-13 No. 3
August 1971
- [3] Paul, H.:
Untersuchungen zur Signalübertragung im Zeit- und Sequenzbereich. Thesis submitted to Institut für Allgemeine Nachrichtentechnik, TH Darmstadt, BRD, December 1969 (unpublished)
- [4] Schmidt, E., Roth, D.:
Sequenz-Multiplex von binären PCM-Signalen bei gestörtem Übertragungskanal.
Int. Elektronische Rundschau
25(1971) pp. 59-63

- [5] Bagdasarjanz, F.:
Sequenz-Multiplexsystem für Daten-
Übertragung.
AEO 25(1971) pp. 579-585
- [6] Harmuth, H.F.:
Transmission of information by
orthogonal functions.
Berlin: Springer-Verlag 1969
- [7] Hübner, H.:
Methods for Multiplexing and
Transmitting Signals by means of
Walsh Functions.
Proceedings of the 1971 Symposium
on Theory and Applications of
Walsh Functions.
Hatfield, England, June 1971
- [8] Gordon, J.A., Barrett, K.:
Correlation-recovered adaptive
majority multiplexing.
Proc. IEE Vol. 118 No. 3/4
March/April 1971, pp. 417-422
- [9] Oberhaus, E.R.:
Untersuchung eines Sequenz-Mul-
tiplex-Prinzips für binäre Signale
mit Mehrheitsentscheidung.
Thesis submitted to Institut für
allgemeine Nachrichtentechnik,
TH Darmstadt, BRD, June 1971
(unpublished)
- [10] Harmuth, H.F., Frank, T.H.:
Multiplexing of digital signals
for time-division channels by
means of Walsh Functions.
Proceedings of the 1971 Symposium
on Theory and Applications of
Walsh Functions.
Hatfield, England, June 1971
- [11] Vary, P.:
Untersuchung eines Sequenz-
Multiplex-Prinzips für binäre
Signale mit dyadischer Ver-
knüpfung.
Thesis submitted to Institut für
Allgemeine Nachrichtentechnik,
TH Darmstadt, BRD, December 1971
(unpublished)

AN ADAPTIVE DIGITAL VOICE MULTIPLEXER USING WALSH FUNCTIONS

H. E. Jones

Westinghouse Electric Corporation

Baltimore, Maryland

Introduction

The transmission efficiency of multiplexed digital voice channels can be significantly improved if the distribution of the total channel capacity among the various baseband channels is not fixed but is varied on the basis of the instantaneous requirements. The reason for this is that the amplitude of a single voice signal varies from relatively short periods of high levels to long periods of very low levels. While signal companding (attenuating the higher signal levels at the encoder input and the lower levels at the decoder output - compressing and expanding) can improve the situation there still remain the conversational pauses. Brady⁽¹⁾ has obtained data which indicates an activity factor of only 40% for one-way in a two-way conversation. Thus a coding scheme based on an equally spaced sample values taken on a single voice signal will produce the zero-level code word with a probability of 0.6. This in itself leads to a fairly poor transmission efficiency even when only a small number of distinct code words are used.

An indication of the extent of this inefficiency can be obtained from the data reported in either Brady⁽¹⁾ and (2) or in Sunde⁽³⁾. If we establish the criterion that 99% of the voice signals are to be below the maximum coding level at least 99% of the time, Brady's data indicates that the mean power level of a channel must be, on the average, about 26 dB below the maximum coding power level while Sunde's data shows this to be about 28 dB. Of this, 4 dB represents the 60% inactivity. In comparison, the same criterion could be satisfied with an average mean power level only 8 dB below the maximum coding power level if the signal amplitude were normally distributed. This 18 or 20 dB is a significant difference. If, for example, a 20 dB improvement in signal-to-noise ratio were obtainable in a linear PCM encoder, the number of bits per sample needed could be reduced from seven to four or from six to three.

In simultaneously encoding a number of voice channels, these inefficiencies can be measurably reduced by encoding the several baseband signals collectively. In doing this, advantage can be taken of the statistics of the com-

posite. From the central limit theorem, the amplitude distribution of the sum of a number of voice signals approaches a normal distribution as the number becomes large, so even though each channel might require a peak-to-mean coding range of 28 dB the peak-to-mean coding range of the overall encoder may only have to be somewhat greater than the 8 dB needed for a normal distribution. This can only be obtained if the proportioning of the overall coding capacity among the baseband channels is not predetermined but rather is allowed to vary in accordance with the instantaneous demands. In this way, the capacity released by one baseband channel during periods of inactivity can be utilized for the transmission of information pertaining to the other channels.

Besides having a code that permits the total channel capacity to be proportioned among the individual baseband channels as needed, the encoder/multiplexer will also require some means of determining what this proportioning should be. This should, of course, be whatever is required at the time to achieve the best performance from the equipment, which implies that the encoder/multiplexer must be capable of judging how badly things are going so that it can juggle the channel capacities around to try to make things better. One version of this technique is used in multi-channel Delta modulation.

The Multi-Channel Delta Modulator

A block diagram of the multi-channel Delta modulator is shown in Figure 1. In this device the performance of the encoding process is measured by comparing the input signals, $S_i(t)$, with what the modulator believes the demodulator outputs to be. These estimates of the recovered signals, $\hat{S}_i(t)$, are obtained by feeding the output binit stream into a model of the demodulator. The differences between these signal-pairs are fed through linear "weighting" filters, multiplied by one of a set of Walsh functions, $\Phi_i(t + \tau)$, and summed to form the composite error signal. While the magnitude of this error signal represents a measure of how badly the modulator is perform-

ing, little of this utilized in this rather simple encoder. Rather, in a manner similar to that used in Delta modulation (hence the name), the modulator attempts to maximize the performance by selecting at each clock time a pulse of whichever polarity is required to drive the composite error towards zero.

The weighting filters are used to construct a reasonable performance measure from the $S_i(t) - \hat{S}_i(t)$ differences. The weighting used is integration plus an instantaneous component (lead) added to obtain good rebalancing behaviour. The transfer functions of these weighting filters are proportional to:

$$W(S) = 1 + \frac{\Omega}{S} \quad (1a)$$

$$\text{where } \Omega = 3 \times 10^3 (\text{seconds})^{-1} \quad (1b)$$

The reconstructed baseband signal estimates are obtained by passing the output binit stream, multiplied by the same Walsh functions delayed one clock time, through linear predicting filters. These filters are used to obtain the minimum variance linear estimates of the baseband signals at each clock time based on the preceding multiplexer output. This minimum variance requirement can be achieved if the impulse response of the filter is proportional to the autocorrelation function of voice signals for $t > 0$. The filter used has a transfer function proportional to

$$F(S) = \frac{1}{S + \Omega} \quad (2)$$

The impulse response of this filter,

$$h(t) = \begin{cases} \exp[-\Omega t] & \text{for } t \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

is a reasonably close approximation to the desired autocorrelation function which was obtained from the power spectrum of composite speech reported by Tarnoczy (4).

The set of Walsh functions are used to provide the variable proportioning of transmission capacity among the several basebands. From a geometric viewpoint, these functions can be considered as elements in an L -dimensional linear vector space (under modulo-2 addition), where LT is the length of the longest period of the functions and T is the clock period. If the N functions used are linearly independent, they span an N -dimension linear manifold of this L -space.

The output binit stream can be taken L binitis at a time, resulting in a sequence of elements, $\{s_i\}$, in the L -space (all possible amplitude transitions of the output stream and the Walsh functions occur at clock times).

Define the inner product:

$$(\alpha_1, \alpha_2) \triangleq \int_0^{LT} \exp[\Omega(t-LT)] \alpha_1(t) \alpha_2(t) dt \quad (4)$$

The output from a predicting filter can be obtained by convolving the input with the impulse response given in (3). Using the inner product defined in (4), the output from the K^{th} filter at the end of the β_1 binitis can be expressed as:

$$\hat{S}_{k,1} = \hat{S}_{k,1-1} \exp[-\Omega LT] + (\phi_k, \beta_1) \quad (5)$$

The first term on the right represents that part of the output value based on all output binitis preceding β_1 while the inner product represents that portion of the output due to the β_1 binitis. The baseband signals will be effectively separated if the ϕ_i 's are orthogonal with respect to this inner product. In general, ΩLT is much less than unity, the kernel in (4) is approximately unity and the Walsh functions are nearly orthogonal.

Computational Model

The behaviour of the Multi-Channel Delta modulator can be computed from the discrete-time model shown in Figure 2. In this model, the signals at various places in the multiplexer are represented by real-valued sequences corresponding to sample values taken on the continuous signals at clock times. The model uses only one of the N channels with the effects of the remaining channels being introduced by the sequence $\{n_i\}$. To provide some continuity with the preceding discussions, the baseband input signal is represented by $\{s_i\}$. The subscript will now be used to indicate a particular member of the sequence rather than a channel, which it signified in $S_i(t)$. This is no longer needed since this section is concerned with the model of Figure 2, which has only one identifiable channel.

The recursive relationships among the sequences are:

$$e_n = s_n - \hat{s}_n \quad (6a)$$

$$\hat{s}_n = b_n + K \hat{s}_{n-1} \quad (6b)$$

$$x_n = e_n - K e_{n-1} + x_{n-1} \quad (6c)$$

$$b_n = \text{Sgn}[x_{n-1} - n_{n-1}] \quad (6d)$$

where

$$K = \exp[-\Omega T] \quad (6e)$$

Since $b_n = \pm 1$, the peak encoded power level is, using (6b),

$$\rho = \text{Max}_n (s_n^2) = \frac{1}{(1-K^2)} \quad (7)$$

With the number of channels, N , sufficiently large, the amplitude distribution of the sum of the baseband signals is roughly normal, in which case 99% of the channels will overload less than 99% of the time if the variance of the

composite is at least 8 dB below ρ . Thus the variance of $\{s_i\}$ is taken to be

$$\sigma_s^2 \triangleq \langle \{s_i^2\} \rangle = \frac{.15}{N} \rho = \frac{.15}{N(1-K^2)} \quad (8)$$

Define the quantization error sequence $\{\epsilon_i\}$, with

$$\epsilon_n \triangleq b_n - x_{n-1} \quad (9)$$

From equations (6) and (9):

$$x_n = s_n - K s_{n-1} - \epsilon_n \quad (10)$$

The variance of $\{x_i\}$, using equations (9) and (10) and recalling that $\langle \{s_i s_{i-1}\} \rangle \approx K \sigma_s^2$, is:

$$\sigma_x^2 \triangleq \langle \{x_i^2\} \rangle = (1 - K^2) \sigma_s^2 + \sigma_\epsilon^2 \quad (11a)$$

where

$$\sigma_\epsilon^2 \triangleq 2 \langle \{b_i(s_i - K s_{i-1})\} \rangle + \langle \{\epsilon_i^2\} \rangle \quad (11b)$$

Thus $\{x_i\}$ can be viewed as the sum of two uncorrelated sequences, one, with variance $(1 - K^2) \sigma_s^2$ representing the prediction error and the other, with variance σ_ϵ^2 , the quantization error.

The baseband input signals are statistically independent and because of the Walsh function multiplication, the error signals are uncorrelated. So the variance of $\{n_i\}$ is just $(N - 1)$ times the variance of $\{x_i\}$. So

$$\sigma_n^2 = (N - 1) [(1 - K^2) \sigma_s^2 + \sigma_\epsilon^2] \quad (12)$$

For these same reasons, the elements of $\{n_i\}$ are uncorrelated and, in addition, $\{n_i\}$ is of zero mean.

Computation of Quantization Noise

The baseband quantization noise for various numbers of channels was computed using the computational model and a post-detection frequency band from 200 Hz to 4 kHz. Since the computer program was not, at the time, capable of computing the amplitude distribution for the composite baseband signal, a normal distribution was assumed even though the numbers of channels used were so small. In addition, the first term in the quantization noise of (11b) was assumed to be negligibly small. This assumption should hold reasonably well since $(s_i - K s_{i-1})$ is small at the bit rates used and b_i is not too strongly dependent on $(s_i - K s_{i-1})$ because of $\{n_i\}$.

The computational procedure used was to assume an initial value for σ_ϵ^2 . This value is then used in (12) to obtain σ_n^2 . The assumption that $\sigma_\epsilon^2 = \langle \{\epsilon_i^2\} \rangle$ leads to a discrete Markov process and the transitional and approximate state probabilities can be obtained from σ_n^2 and the assumed normal distribution. A new value for σ_ϵ^2 is computed from the state

probabilities and the procedure is repeated, converging to a solution. The transitional and state probabilities used in obtaining the final value for σ_ϵ^2 are then used to compute the autocorrelation function of $\{x_i\}$, which in turn is used to compute the power spectrum of $\{e_i\}$, or, to be more accurate, the integral of the power spectrum over the band assumed for the post-detection filter.

The results of this are shown in Figure 3 along with actual data obtained from a five channel breadboard and the theoretical linear PCM curve for comparative purposes. The "peak signal" was taken to be $0.5(1-K)^{-2}$, which is the RMS value of the maximum monochromatic tone that can be encoded without amplitude limiting.

Concluding Remarks

As can be seen in Figure 3, the signal-to-quantization noise is reasonably good in spite of the simplicity of Multi-Channel Delta modulation. The modulator also offers other desirable features. One of these is that standby channels can be readily provided since the additional circuitry needed is minimal. This is due partly to the Walsh function modulation used to separate the basebands and partly to the adaptive behaviour. The main effect in activating stand-by channels is an increase in the quantization noise.

The spectrum of the quantization noise is very smooth with a maximum occurring at from one to three kHz depending on the clock rate. The coloration is barely noticeable aurally, and, in fact, most listeners take the noise to be white.

Another feature of the modulator is that the sensitivity to transmission bit errors is low, with a transmission error appearing as a single one-pulse error on all the channels. This also is a consequence of using Walsh functions to separate the channels. In comparison, a single transmission error in PCM/TDM would hit only one channel but the effect on this one channel would be, on the average, much greater. This would also be the case if address preambles were used in the adaptive multiplexer, but there would also exist the possibility of error occurring in the preamble causing the correct code word to be directed to the wrong channel.

As for further improvements, our interest at this time is directed mainly to the weighting functions. Presently, linear weighting is used, with the result that errors in a channel with a high baseband signal level are weighted equally with those in a channel with a low base-

band signal level. We intend to improve on this by introducing a weighting function scale factor inversely proportional to the input signal level. Since the baseband signal is not directly subjected to this non-linearity the demultiplexer is unchanged and it follows that the transmission bit error sensitivity would not be increased.

Acknowledgment

The author wishes to acknowledge the encouragement and suggestions offered by Dr. B. A. Shenoi of the University of Min-

nesota and by J. C. G. Carter, R. H. Donaldson and Dr. R. T. Hilbish of the Westinghouse Electric Corporation.

References

1. P. T. Brady, "A statistical analysis of on-off patterns in 16 conversations", BSTJ, Vol. 47, Jan. 1968, pp 73-91.
2. P. T. Brady, "A statistical basis for objective measurement of speech levels", BSTJ, Vol. 44, Sept. 1965, pp 1453-1486
3. E. D. Sunde, Communications Engineering Theory, Wiley, New York, 1969, pp 63-66
4. T. Tarnoczy, "Détermination du spectre de la parole avec une méthode nouvelle", Acustica, Vol. 8, 1958, pp 392-395

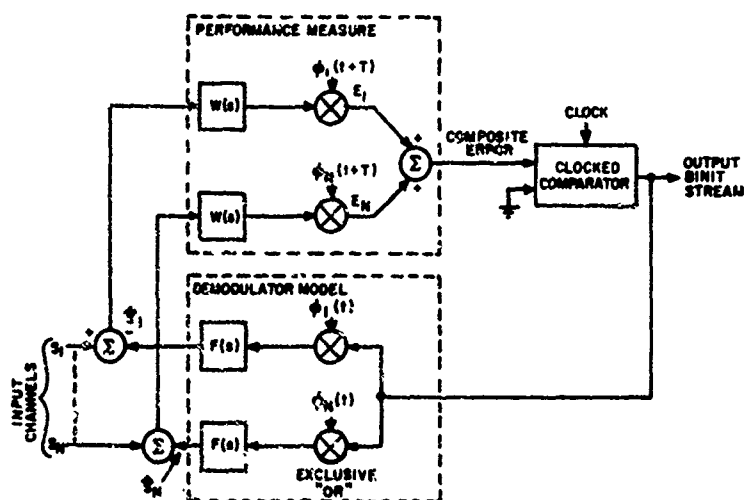


Figure 1: Multi-Channel Delta Modulation

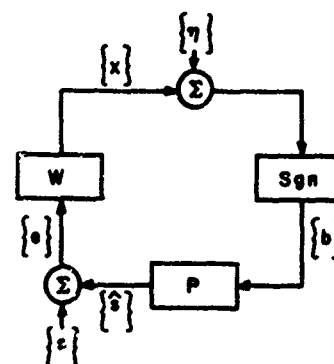
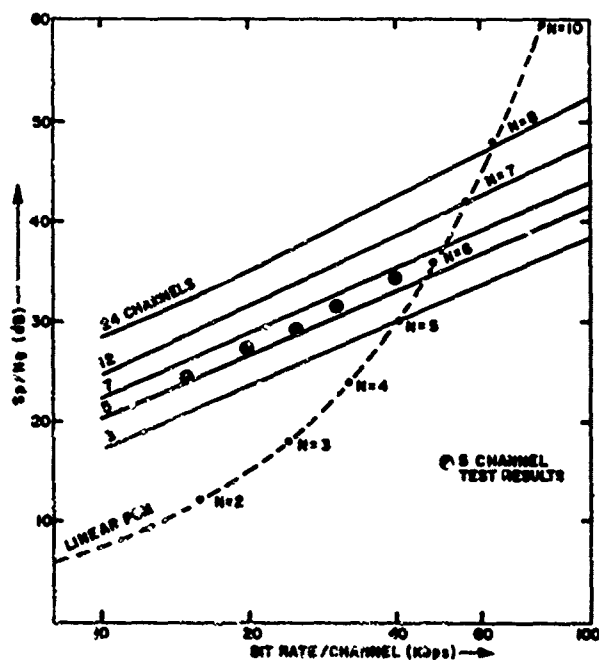


Figure 2: Computational Model

Figure 3: Peak Signal-to-Quantization Noise



WALSH FUNCTION GENERATOR FOR LABORATORY USE

Roque Lopez de Zavalía
Enzo S. M. Moro
Electrical Engineering Institute
Tucuman National University, Argentina

Translated by Eduardo Schonborn, Jr.
DCA System Engineering Facility
Reston, Virginia 22070

Summary

Utilizing the Rademacher functions provided by a bistable chain, it is possible to simultaneously synthesize 20 Walsh functions, whose normalized and non-normalized sequence can be varied. The device also provides command pulses for low-pass sequence filters and an arrangement for synchronization with another generator is included.

Introduction

Upon confronting the experimental study in the practical possibilities of utilizing Walsh functions [1, 2] in electronics, the first problem faced by the designer is that of disposing of the necessary instruments. The measurements obtained from the instruments will make possible the comparison between the theoretical results and those provided by the practical application with its inherent natural limitations.

The applications of the Walsh functions in practical electronics is actually found in the typical initial development stage in the laboratory. The instruments needed for the specific measurements are not offered commercially, while on the other hand the practical information given by publications is scarce. Logically a good part of the available instruments in an electronics development laboratory finds application when one is working with Walsh functions, with the exception of certain instruments conceived expressly for sinusoidal signals; for example, generators and filters for these functions.

An experimental study plan regarding Walsh function was started at the Electrical Engineering Institute of the Tucuman National University. The first problem that was presented was, naturally, that of disposing of a Walsh functions generator. In a previous development a generator had been obtained that provided 20 functions and command pulses for low-pass sequence filters. With the same generator a sequence multiple for three telephone channels was attempted [3]. With the experience gained in that first model, the second version of the instrument was prepared as described in this current article.

The second model differs from the first, essentially, in the inclusion of certain characteristics that make it possible for it to be utilized as a generator fixed in sequence by an external signal. A previous article [4] treated the theoretical aspects of the synthesis of Walsh functions in electronics, while this article only considers the practical points corresponding to the development and realization of a generator for laboratory use.

1. Block Schematic

In figure 1a the distinct sections of the device are represented in a functional schematic:

1. Trigger pulses section - generates the pulses to trigger sections 2 and 3. Permits varying the non-normalized sequence from 15 zeros per second to 16,000 zeros per second. It can operate as:

a. Free running oscillator.

b. Fixed oscillator; in which case the normalized sequence $i = 1$ is fixed in sequence with an external signal.

2. Sequence division section - permits the normalized sequence i to be divided by the factors 1, 2, 4, and 8.

3. Walsh function synthesis section - generates simultaneously 10 odd functions and 10 even functions with normalized sequence $i = 1, 2, 3, \dots, 10$.

4. Command pulses section - its objective is to generate two pulses that will permit the command of an external circuit (low-pass sequence filter).

To simplify the description of the device it is convenient to discuss first a fundamental model that consists only of the Walsh function synthesis section and the oscillator section (Figure 1b). A later step will consider the aggregate of the sequence division section and the command pulse generator section. Therefore, that which follows will use this order for discussion.

II. Walsh Function Synthesis Section

The adopted [1] scheme of generation indicated in Figure 2 permits obtaining 10 odd functions, $\text{sal}(1, \theta)$, and 10 even functions, $\text{cal}(1, \theta)$, with normalized sequency $i = 1, 2, \dots, 10$.

The bistable chain (bistables B_0 to B_5) delivers $\text{sal}(2^k, \theta) + A \text{cal}(0, \theta)$ by having utilized bistables whose voltage vary between a value near zero and some positive voltage. It is convenient to note that if one supposes that an output of one bistable (i.e. B_k) delivers one voltage $A \text{cal}(0, \theta) + \text{sal}(2^k, \theta)$ the complementary output B_k will deliver the voltage $A \text{cal}(0, \theta) - \text{sal}(2^k, \theta)$.

Starting from the Rademacher functions $\text{sal}(2^k, \theta)$ it is possible to synthesize the Walsh functions effecting products of Rademacher functions. Therefore, it is necessary to accomplish electronically the product operation of normalized Walsh functions, of which $\text{sal}(2^k, \theta)$ are a particular case.

A Walsh function takes only +1 or -1 values. The product of two functions will therefore be +1 if both are of the same sign in a certain time; and if that condition is not met it will be -1. The factor functions as well as the product function can take only two well defined values, which makes it possible to utilize binary algebra. Let the product of two functions be W_1 and W_2 . For them we can make the following truth tables:

(a)			(b)			(c)		
W_1	W_2	W_a	W_1	W_2	W_b	W_1	W_2	W_c
+	+	+	1	1	1	0	0	0
-	-	+	0	0	1	1	1	0
+	-	-	1	0	0	0	1	1
-	+	-	0	1	0	1	0	1

Table (a) refers to the signs of the product of two Walsh functions, table (b) to the corresponding logic function if a +1 (Walsh function) is taken as equivalent to a logic 1, and a -1 equivalent to a logic 0, while table (c) indicates the logic function if a -1 is taken as equivalent to a logic 1 and +1 equivalent to a logic 0. For tables (b) and (c) one has:

$$(b) \quad W_b = W_1 W_2 + \bar{W}_1 \bar{W}_2$$

$$(c) \quad W_c = W_1 \bar{W}_2 + \bar{W}_1 W_2$$

These expressions are implemented with digital gates; both variables require disposing of the binary variable and its complement (negation). It is opportune to note that while the (b) expression provides the desired Walsh function, expression (c) also provides it, but delivers the function with a changed sign. Strictly speaking, the out-

put of the gates is equal to a positive constant plus the product function (case b) or less the product function (case c).

In figure 2 are indicated in symbolic form (the circles represent the product operation) the operations that must be implemented to obtain each function, for which a unique standard circuit is utilized, excited just as inferred by the expression (b) or (c). The utilization of one or other variable is a question of practical convenience, the only detail that must be observed in essence is the fact that the function must appear in the series with the correct sign.

Between the direct outputs of the bistables and the gates, one has 20 Walsh functions. These signals are taken to 20 separating amplifiers designed in such a form that an eventual short circuit at the output of the amplifiers does not produce any harm neither in the amplifiers nor in the integrated TTL circuits utilized in the logic (bistables Texas SN5474 and gates Texas SN5450 and SN5400).

Prior to passing the signals through the separating amplifiers, the 20 Walsh functions are taken to a connector situated at the rear panel of the device. In the front panel there are two keys of 10 positions each, Channel A and Channel B. Each key allows to choose simultaneously the even function and the odd function for a given sequency ($i = 1, 2, \dots, 10$), with outputs for the function with the correct sign or with the changed sign. In addition, the signal $-\text{sal}(1, \theta)$ is taken to a BNC connector on the front panel, to be utilized as a synchronizing signal for an oscilloscope.

III. Trigger Pulses Section

Figure 3 presents the schematic of the principle of this section. The rectangle marked (1) symbolizes an oscillator controlled by voltage. It is of the LC type, utilizing a variable capacitance diode to obtain fine frequency adjustment through the action of a control voltage coming from the phase comparator (4) and the low-pass filter (3). By means of a variable capacitor, whose axis is controlled from the front panel of the device, it is possible to vary the frequency of the oscillator between 480 KHz and 1100 KHz. The circuits described permits two modes of operation: a) as a free oscillator, and b) as an oscillator controlled by voltage. If one desires, it is possible to use an external oscillator as a substitute.

It is convenient to consider first the case of the free running oscillator (or using an external oscillator). The signals from the oscillator (internal or external) pass through block (2) which is a separating amplifier and limiter; and furnish trigger

pulses to the chain of 9 bistables (Texas SN5474) indicated symbolically as $B_{17}, \dots, B_{13}, \dots, B_9$.

The output of circuit (2) and the outputs of the 9 bistables are taken to a 10 position switch L_p , that permits selecting signals whose frequency is smaller as the switch moves to the bistables of the lower subscripts.

It has been stated that as a first step it would be assumed that the trigger pulses section would be connected directly (Fig 1b) to the synthesis section. This mode of operation presents itself in one of the applications of the device; i.e. when the frequency division factor is equal to 1 (see paragraph 5). In this case, between the synthesis section (6 bistables) and the trigger pulses section (9 bistables) one can dispose of a total of 15 bistables. If all the bistables of this last section are utilized, the frequency of the signal $sal(1, 0)$ provided by the B_0 bistable will be $1/2^{15}$ of the frequency of the oscillator; that is, it will be possible to vary the $sal(1, 0)$ frequency approximately between 15 zeros per second to 33 zeros per second.

Being bothersome to work with division factors, it was chosen to assign to the oscillator (front panel) a nominal frequency of 15 to 30 zeros per second and to mark the L_p switch (front panel) with multiplication factors $2^0, 2^1, \dots, 2^9$, as it is shown on Figure 3. The $sal(1, 0)$ signal provided by the B_0 bistable is taken to a BNC connector located on the rear panel, which permits the utilization of a digital frequency-meter to determine (if it was necessary) the non-normalized frequency of $sal(1, 0)$, and which in conclusion, can be varied between 15 zeros per second and slightly more than 16,000 zeros per second. If an external oscillator is used, the signal from B_0 can have as low a frequency as desired, and as high as the bistables utilized will permit.

Assuming always the connection of the Figure 1b, the following will discuss the operation of the oscillator when it is synchronized by an external signal (Figure 3). For this operation it is necessary to take the synchronizing signal and the $-sal(1, 0)$ signal provided by the B_0 bistable (synthesis section) to the two inputs of a bistable (KTL9923) that acts as a phase comparator (4). To be capable of triggering the bistable, the signal from B_0 must be previously differentiated by means of circuit (5); while the external signal can first pass through a good limiter amplifier and then be differentiated; operations which correspond to block (6) of Figure 3.

The aggregate of trigger pulses and synthesis sections operate as a phase-locked

loop*; a subject over which there is ample bibliography. In reference [5], that subject is discussed relative to the utilization of a divider chain. In order to operate using a synchronizing signal, the B_0 signal is taken to a frequency close to the fundamental frequency of the external signal. This is accomplished by actuating switch L_p and the variable capacitor of the oscillator to bring the device into a synchronizing capture zone. With a small variation of the variable capacitor it is possible, if one desires, to obtain coincidence of zero crossings between the synchronizing signal and the $sal(1, 0)$ signal; and also, to obtain null time delays between both signals. Switch L_p is in reality a switch with two commutator sections, one of them for the multiplication factor already explained, and the other to commutate capacitors and to vary the RC constant of the low-pass filter (3).

IV. Command Pulses Section

This part of the device has as its objective to control a low-pass frequency filter [1], that is an alien arrangement to the generator per se, and which needs to be synchronized with the generator for its operation in receiving command pulses.

In succinct form the work cycle of a low-pass frequency filter (Figure 5) is the following: The input signal $F(t)$ is integrated during time T ; at the end of the integration interval the output voltage of the integrator is transferred to memory capacitor C_m , for which switch L_y goes from position 1 (position for the integration) to position 2 (transfer). Immediately, the switch goes to position 3 (erased), taking the integrator to its initial conditions to begin a new integration interval (switch to position 1). The commutation of the switch is realized in actual practice with two command pulses.

Let B_1 be the pulse that determines the transfer of the information to the memory capacitor, and let B_2 be the erasure pulse. To minimize the error both operations must be performed in the least possible time, and B_2 must appear immediately after B_1 .

The previous statements are ideal conditions which must be reconciled with those of the practical behavior of the low-pass filter that is to be controlled.

*In an earlier work [4] it was stated that it did not make sense to talk about phase difference for Walsh functions, so that for the present case it would be correct to refer to a frequency-locked loop, and to call the phase comparator (bistable) as a time delay comparator.

The command pulses (Figure 6) are B_1 and B_2 ; and both have a duration (high positive) of $t = T/64$, where T is the integration interval, corresponding to the $\text{sal}(1, \theta)$ periodic interval, just as it is explained further ahead (paragraph 5). The triggering of the B_1 pulse can be realized for $\theta = -1/2$ or $\theta = 0$, selectable by means of switch L (front panel). Pulse B_2 is present immediately after pulse B_1 . Figure 6 includes a schematic of the logic used (bistables Texas SN5474 and gates Texas SN5400 used as inverters, N). Pulses B_1 and B_2 and their complements \bar{B}_1 and \bar{B}_2 are taken through separating amplifiers to the front panel and to the connector located in the rear panel, which is the same connector where the 20 functions from the synthesis section also meet.

V. Normalized Sequence Division Section

In the entire discussion that preceded, it was assumed that bistable B_0 (synthesis section) corresponds to the unitary sequence, providing $\text{sal}(1, \theta)$. Multiplying the oscillator sequence (15 to 30 zeros per second) by the multiplication factor (see paragraph 3) one obtains the non-normalized sequence for the B_0 bistable.

It has been stated that in the low-pass sequence filter the (theoretical) integration interval must correspond to the repetition interval T of the $\text{sal}(1, \theta)$ function.

For the behavioral trial of a low-pass sequence filter it is convenient to dispose of signals with fractional sequence. This is obtained by means of the sequence division section, which is inserted (Figure 1a) between the trigger pulses section and the synthesis section.

The sequence division section simply consists of three bistables (B_6 to B_8) connected as dividers by factors of two. When the device operates with sequence division factor equal to 1 those bistables are not used. If it is desired to operate with sequence division factors of 2, 4, or 8 the necessary bistables are inserted, by means of a switch labeled sequence division factor (front panel). In this case one must keep in mind that the normalized sequence which are indicated by the numbers located together with the knobs of the 10 position switches (channel A and channel B) described in paragraph 2, must be related to the mentioned factor of sequence division. For example, $\text{sal}(7, \theta)$ will correspond to $\text{sal}(7, \theta)$ if it is operating with sequence division factor equal to 1, but it will be $\text{sal}(7/2, \theta)$ if it is operating with division factor of 2.

The previous discussion really has physical meaning only if it is joined with the command pulses for low-pass sequence

filters; for that operation, the switch labeled sequence division factor must also perform the necessary computations so that in every case the command pulses will fulfill the specifications defined in paragraph 4.

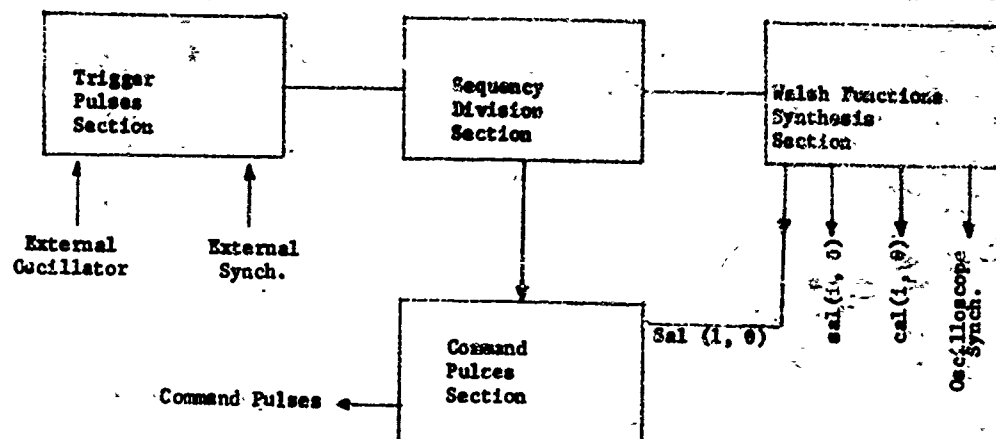
Acknowledgement

The present work has been realized with the economic support of the Tucuman National University and the National Commission of Physical Studies.

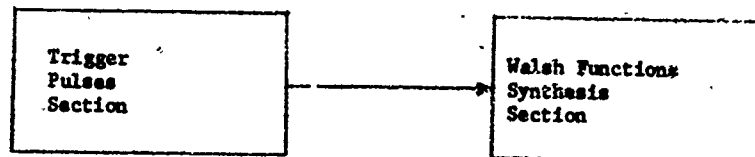
I wish to thank also Texas Instruments Argentina, who performed an important donation upon the Electrical Engineering Institute of the Tucuman National University. The integrated digital circuits used in the device were obtained from this donation of semiconductors.

Bibliography

1. H. F. Harmuth, *Transmission of Information by Orthogonal Functions*, Springer-Verlag, 1969.
2. F. Pilcher, *Synthese Linearer Periodisch Zeitvariabler Filter Mit Vorgeschriebener Sequenzverhalten*, Archiv der Elektrischen Übertragung, 1968, pages 150-161.
3. R. Lopez de Zavalia, E. S. M. Moro - Walsh Functions, Sequence Filters and Sequence Multiples, International Conference Mexico 1971, About Systems, Networks, and Computers January 1971.
4. R. Lopez de Zavalia - The Walsh Functions and Their Generation in Electronics, to be published in Volume No. 10 (1971) of the Electrical Engineering Institute Magazine of the Tucuman National University.
5. R. A. Rivero, A. Carline - Frequency Multiplier Using An Oscillator Circuit Controlled by Phase, Volume No. 9 (1970) Electrical Engineering Institute Magazine of Tucuman National University.
6. Texas Instruments Inc. - Technical Specifications, Solid Circuit, Semiconductor Network, Series 54 (Bulletin No. DL-5.6710107, May 1967).



a) COMPLETE FUNCTIONAL SCHEMATIC



b) SIMPLIFIED FUNCTIONAL SCHEMATIC

FIGURE 1. FUNCTIONAL SCHEMATIC

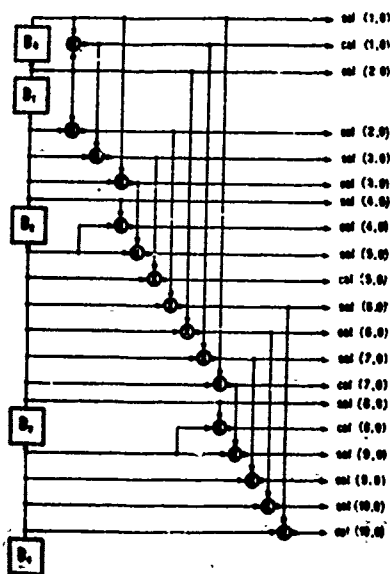


FIGURE 2. WALSH FUNCTION GENERATION

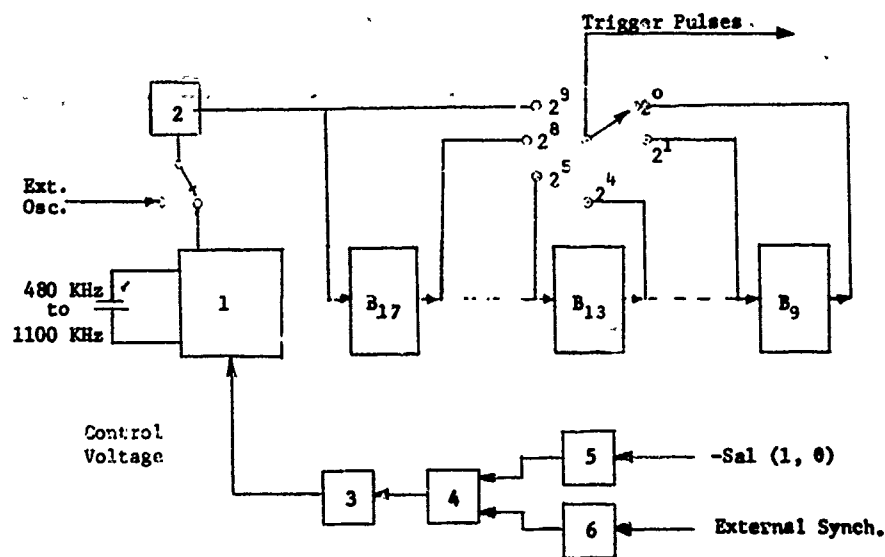


FIGURE 3. TRIGGER PULSES SECTION



FIGURE 4. FRONT PANEL

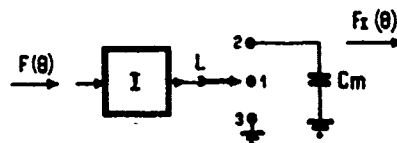


FIGURE 5. LOWPASS SEQUENCE FILTER

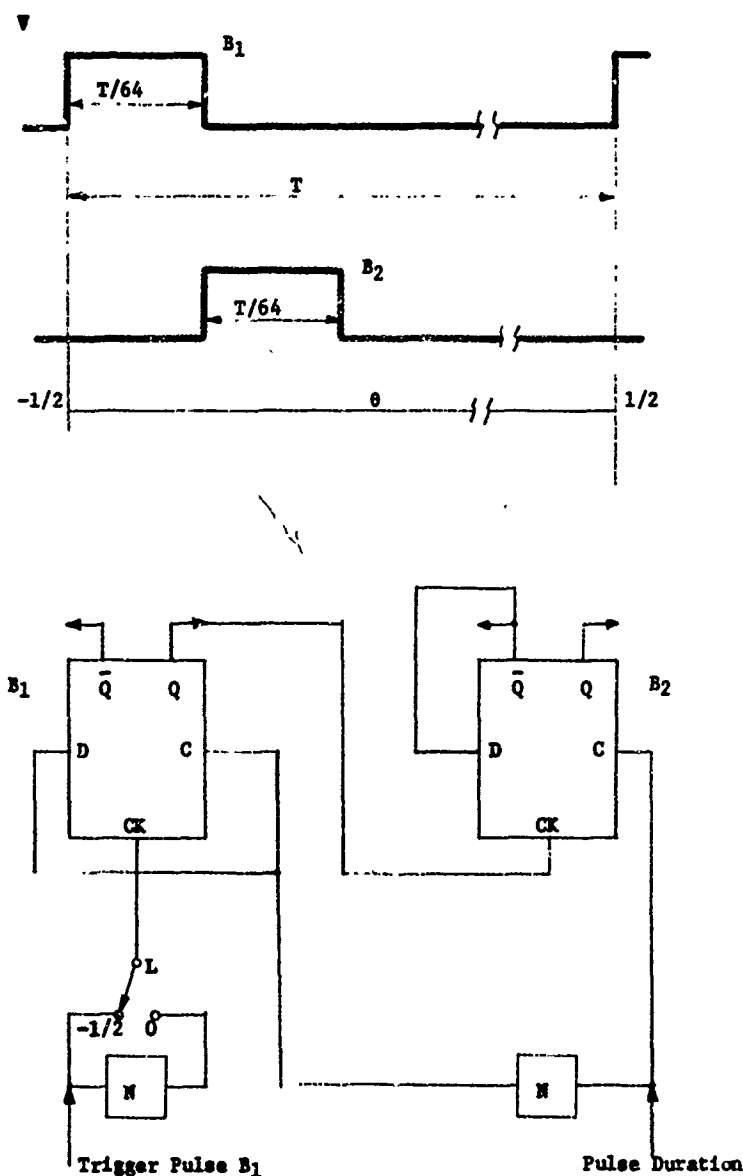


FIGURE 6. COMMAND PULSES

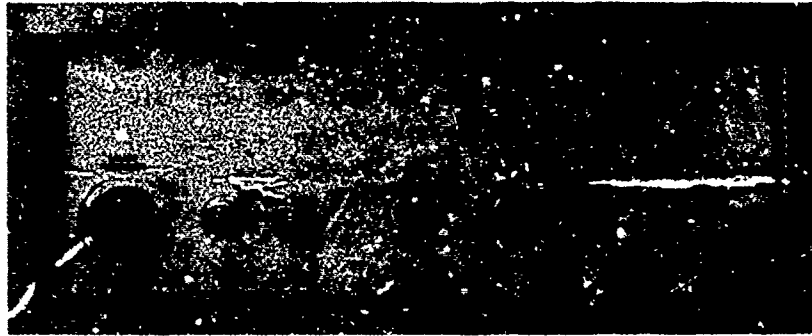


FIGURE 7. REAR PANEL OF THE GENERATOR

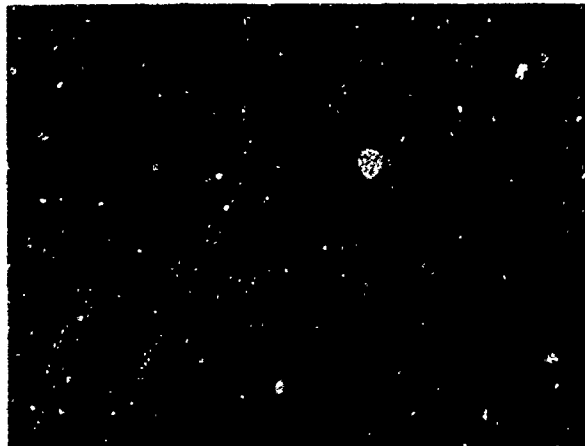


FIGURE 8. GENERATOR VIEW

A NEW WALSH GENERATOR AND ESTIMATION OF THE TOTAL ORTHOGONALITY ERROR OF THESE GENERATORS.

Reiner Nawrath
Institut für Elektrische Nachrichtentechnik
der Technischen Hochschule Aachen
Aachen, West Germany

1. Introduction.

The large number of papers on sequence-technics, which appeared during the last few years, proves the increasing importance of this subject. When realizing sequence systems, as coders or decoders, the development of Walsh-function generators is a special problem. Such generators must comply with the following requirements:

To be capable of producing a large number of Walsh-functions simultaneously and to permit an expansion from m to $2^i \cdot m$ functions.

To effect only a small orthogonality error in the system of generated functions up to high clock frequencies. The binary structure of the Walsh-functions gives reason to develop the generator with digital IC's. To keep the costs low, Standard TTL-IC's should be preferred for the realization.

The orthogonality error is a criterion for the efficiency of generators producing orthogonal functions. It is brought about by three facts: 1. the displacement between two functions caused by propagation delays in IC's, 2. the tolerances of the propagation delay time and 3. the difference between rise and fall times of most IC's.

Varying output voltages of different IC's may be equalized by operational amplifiers and therefore don't affect the orthogonality error.

2. Walsh-function generators and their properties.

One of the possible arrangements of Walsh-functions in the normalized time interval $-1/2 \leq \theta < 1/2$ resulting from the difference equation of Harmuth [1] is shown in figure 1. The same figure shows Walsh-functions in the normalized time interval $0 \leq \theta < 1$ (θ is the time normalized on the orthogonality interval length T). Two of the generators later discussed may be switched over from one time base to the other.

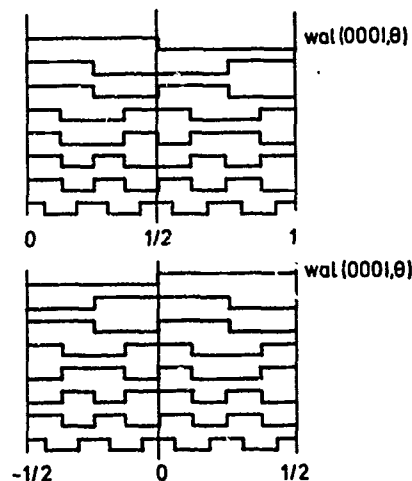


Fig. 1: Arrangement of Walsh-functions for two time bases

2.1 Hitherto existing Walsh-function generators.

In 1968 Harmuth [2] proposed a generator for Walsh-functions, which is reported in practical use. The idea of his concept is based on the multiplication theorem of Walsh-functions.

$$\text{wal}(i, \theta) \cdot \text{wal}(j, \theta) = \text{wal}(i \oplus j, \theta)$$

\oplus denotes the bit per bit mod 2 addition

For generating 2^n Walsh-functions he first produces n Rademacher functions as output waveforms of a n -stage binary flip-flop counter. The remaining $2^n - (n+1)$ functions are obtained by taking all possible mod 2 sums of the digital Rademacher functions:

$$2^n = 1 + \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{n}$$

The 1 denotes the Walsh-function $\text{wal}(0, \theta)$ and $\binom{n}{i}$ specifies all possible combinations of i elements out of n

without repetition. Figure 2 shows the realization of the "Harmuth"-generator with counter flip-flops and halfadders for $n = 4$.

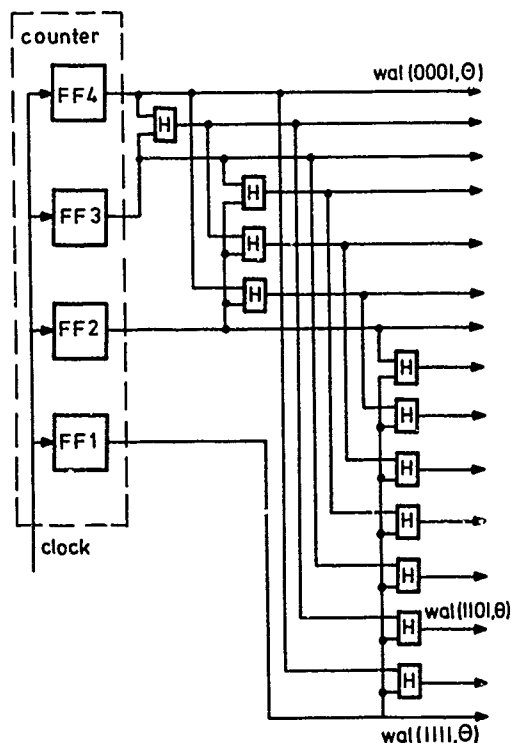


Fig. 2: "Harmuth"-Walsh generator ($n=4$)

Disadvantages of this generator are: The displacement of the Rademacher functions $wal(1,0)$ and $wal(2^n-1,0)$ amounts to $\tau_1 = (n-1)\tau_z$ if an asynchronous counter is chosen (τ_z is the propagation delay of a counter flip-flop). Some other Walsh-functions are further delayed up to $(n-1)\tau_H$ while passing halfadders (τ_H propagation delay of a halfadder). If the asynchronous counter is replaced by a synchronous counter, the maximal time lag between two functions is reduced to $\tau_2 = \Delta\tau_z + (n-1)\tau_H$ ($\Delta\tau_z$ is the difference of the risetimes of two counter flip-flops). Maximal clock frequencies under worst case conditions are $f_1 = 1/(\tau_z + \tau_H) \approx 5$ Mc for a generator with asynchronous counter ($n=4$) and $f_2 = 1/(\Delta\tau_z + 3\tau_H) \approx 10$ Mc for a generator with synchronous counter ($n=4$) respectively.

In 1970 Lebert [3] presented a generator producing a large number of different sequences. It is also possible to change the time base; see figure 3.

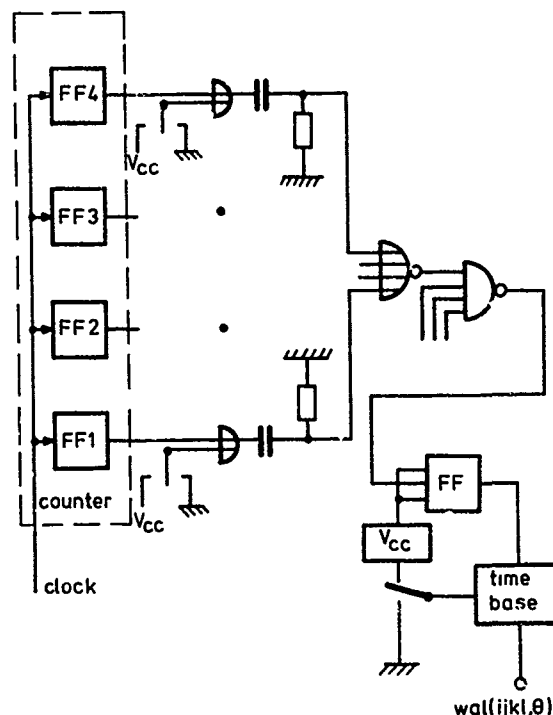


Fig. 3: "Lebert"-Walsh generator ($n=4$)

The elaborateness of digital IC's increases rapidly if one wants to generate more than one function simultaneously. For 2^n Walsh-functions the whole circuit except the counter, must be duplicated 2^n-2 times. In Lebert's design the negative edges of the Rademacher functions produce triggering pulses when passing through differentiators. A particular amount of these pulses is combined by OR-gates and used to trigger an output flip-flop. The maximal time lag between two functions is given by $\tau_3 = (n-1)\tau_z + \Delta\tau_{OR} + \Delta\tau_{RC} + \Delta\tau_{NOR} + \Delta\tau_{FF}$ or $\tau_4 = \Delta\tau_z + \Delta\tau_{OR} + \Delta\tau_{RC} + \Delta\tau_{NOR} + \Delta\tau_{FF}$ dependent from the chosen counter ($\Delta\tau_{OR}$, $\Delta\tau_{RC}$, $\Delta\tau_{NOR}$, $\Delta\tau_{FF}$ are propagation delay time differences of two OR-gates, two differentiators, two NOR-gates and two flip-flops). Fur-

ther delays in the time base have not been considered. Maximal worst case clock frequencies of the "Lebert" generator are $f_3 = 1/\tau_3 \approx 7 \text{ Mc}$ ($n=4$) and $f_4 = 1/\tau_4 \approx 15 \text{ Mc}$ ($n=4$).

A Walsh-function generator developed by Yuen [4] is shown in figure 4. The circuit is a modified version of the "Lebert" generator. The differentiators are replaced by AND-gates which also have the task to synchronize function generation by clock pulses. Furthermore Yuen uses a special counter. The maximal displacement between two produced Walsh-functions is given by $\tau_5 = \Delta\tau_{\text{NAND}} + \tau_{\text{OR}} + \tau_{\text{FF}}$. It must be

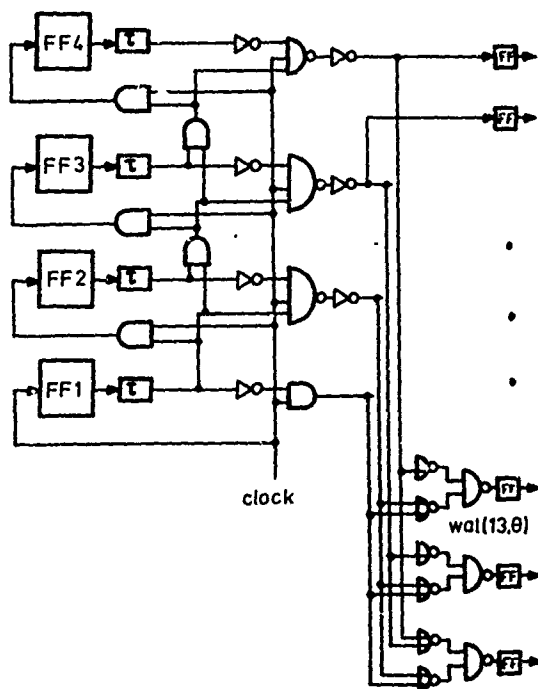


Fig. 4: "Yuen"-Walsh generator ($n=4$)

noted that the propagation delay time of the OR-gate must be calculated from the sum of the propagation delays of a NOR- and a NAND-gate, as OR-gates with four inputs are not available in Standard TTL. The generator can be driven under worst case conditions with the clock frequency $f_5 = 1/(2d + \tau_{\text{FF}} + (n-2)\tau_{\text{AND}}) \approx 12 \text{ Mc}$ ($n=4$) (d is the length of the clock pulse). This frequency is deter-

mined by the counter alone.

2.2 New Walsh-function generator using multiplexers.

In this paper we outline the design of a Walsh-function generator consisting only of counter flip-flops and multiplexers. For producing 2^n Walsh-functions one needs 2^{n-1} multiplexers. Each of the $2^n/k$ inputs of a multiplexer is switched k -times to the output in each interval T . Some input signals must be changed in their logical value with the frequency $f = 2^{(v-n)} \cdot f_c$ if $v \neq 0$. Further one needs $n-v$ counter flip-flops and μ additional gates.

$$k=2^v; v=0,1,\dots,n; \mu = \begin{cases} 0 & v \leq 1 \\ 2^{v-1}-1 & v > 1 \end{cases}$$

Later we shall evaluate which values for v are significant.

The structure of this generator is shown in figure 5 for $n=4$ and $v=0$.

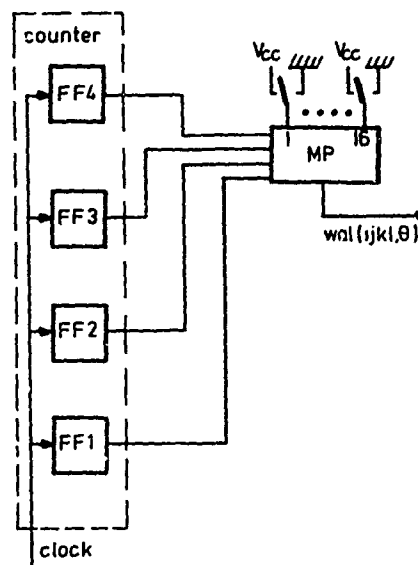


Fig. 5: "Multiplexer"-Walsh generator ($n=4, v=0$)

Incrementing the counter by 1 at each clock pulse one multiplexer input after the other is switched to the output. The desired time base is obtained by choice of the input signals. In the same manner all 2^n Walsh-functions can be generated by one multiplexer in chronological sequence. Attention must be paid to the fact that some multiplexers

invert the input signals.

A simultaneous generation of $m < 2^n = 16$ Walsh-functions (figure 5) is accomplished by connecting in parallel $m-1$ multiplexers with 16 inputs. The maximal time lag between two Walsh-functions comes to $\tau_6 = (n-1)\tau_z + \Delta\tau_M$ when operating with an asynchronous counter and to $\tau_7 = \Delta\tau_z + \Delta\tau_M$ when operating with a synchronous counter ($\Delta\tau_M$ is the propagation delay time difference of two multiplexers). Maximal worst case clock frequencies are $f_6 = 1/(3\tau_z + \tau_M) \approx 7.5$ Mc (asynchronous counter and $n=4, v=0$) and $f_7 = 1/(\Delta\tau_z + \tau_M) \approx 20$ Mc (synchronous counter, independent of $n, v=0$).

Now the operation of the "Multiplexer" generator is discussed for $v=1$. Figure 6 shows one possible realization ($n=4$).

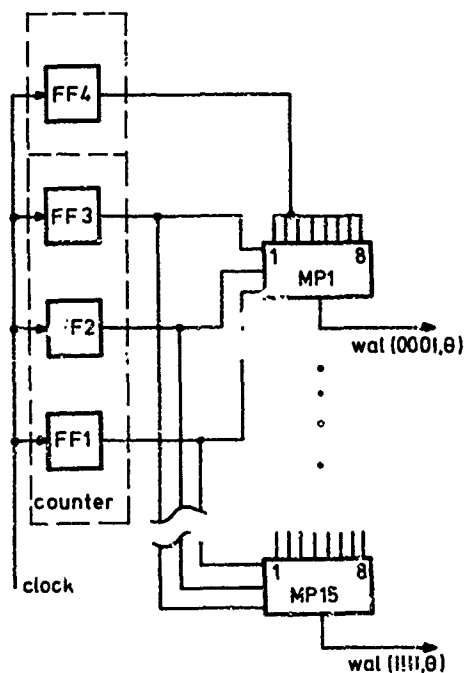


Fig. 6: "Multiplexer"-Walsh generator ($n=4, v=1$)

The 16 line to 1 line multiplexers are replaced by 8 line to 1 line multiplexers. As a 3 bit binary counter is now sufficient one flip-flop may be used otherwise. The counter must run twice from 0 to 7 to get the 16 output levels for one Walsh-function. Each time the counter has reached the decimal eight the input signals of some multiplexers must be changed in their logical level.

Figure 6 shows an example of solving this problem by means of flip-flop FF4. When producing the function $wal(1,0)$ in the normalized interval $-1/2 < \theta < 1/2$ the output signal of the multiplexer must change to logical 1 after $(2m+1)/8$ clock pulses and to logical 0 after $2m+8$ clock pulses ($m=0,1,\dots$). FF4 just changes its output level in this manner and is therefore connected directly to all inputs of the multiplexer MP1. The number of multiplexers whose input signals must be changed is calculated as follows: If the Walsh-function $wal(1,0)$ produced by MP 1 is periodic with $T_n/2$ (T_n is the normalized time interval) then the input signals of MP 1 remain unchanged, otherwise they have to be changed. The normalized sequence [2] of these functions is an even number, and runs up to 2^{n-1} . Therefore the input signals of $b(v=1) = 2^{n-1}$ multiplexers must be changed.

The replacement of the large multiplexers by smaller ones does not bring up any disadvantages. On the contrary advantages are achieved: Maximal clock frequency remains constant, costs and occupied space decrease to 50 %.

Choosing $v=2$ results in the circuit shown in figure 7 ($n=4$). Two of the counter flip-flops are no longer used directly for counting. That is indicated in the figure by separating them from the others by a dashed line.

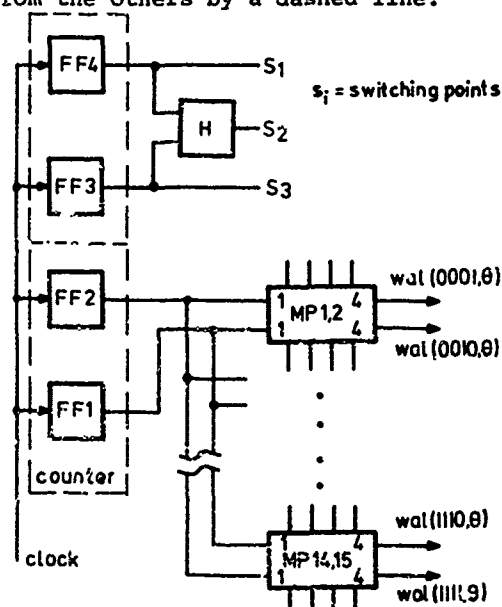


Fig. 7: "Multiplexer"-Walsh generator ($n=4, v=2$)

The number of switching points where some input signals must change its logical values increases from two ($v=1$) to four. These four switching points can't be determined by two counter flip-flops. The requirement of one additional gate arises here. This additional gate, exactly a halfadder, causes a propagation delay time, which decreases the maximal clock frequency. Now it holds $f_8 = 1/(3\tau_z + \tau_M + \tau_H) \approx 6$ Mc (asynchronous counter) and $f_9 = 1/(\tau_M + \tau_H + \Delta\tau_z) \approx 13.5$ Mc (synchronous counter). The costs are still decreasing by this operation as well as the occupied space.

Enhancing v to 3 will result in an increasing number of additional gates and a not desired decrease of the clock frequency below 10 Mc.

Summarizing we state that it is reasonable to build up the generator with a factor $v = 2$ or $v = 1$. The factor $v = 2$ permits to develop generators for 64 different Walsh-functions, using today's available multiplexers (16 line to 1 line) without cascading.

3. Orthogonality error of Walsh generators.

A well known measure to compare the features of different Walsh generators is the orthogonality error of the produced functions. Dealing with logical levels (0 and 1) a possible definition is:

$$\epsilon_{i,j} = \left| 4 \int_{T_n} (\text{wald}(i, \theta) - 1/2) \cdot (\text{wald}(j, \theta) - 1/2) d\theta - \delta_{i,j} \right| \quad (*)$$

where:

$$\text{wald}(i, \theta) := 1/2 (\text{wal}(i, \theta) + 1)$$

$$\delta_{i,j} := \begin{cases} 1 & i=j \\ 0 & i \neq j \end{cases}$$

T_n : normalized orthogonality interval (mainly $T_n=1$)

θ : on T normalized time

Equ. (*) shows, that $\epsilon_{i,j}$ is only different from zero if $i \neq j$. A constant displacement of all 2^n functions against the interval borders does not affect the orthogonality error.

Circuits of Walsh-function generators may be decomposed into three separate stages, see figure 8. First a counter stage - or modified counter stage (Yuen) - second a stage where logical operations are performed and third a stage with output gates.

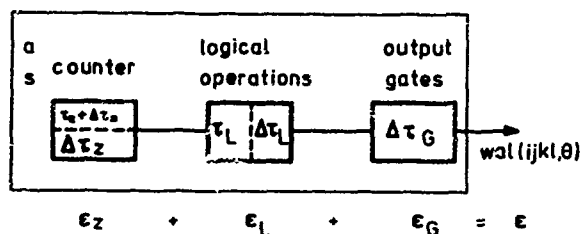


Fig. 8: Block diagram of a Walsh generator and orthogonality error

Analog to the decomposition of the generator the orthogonality error is calculated in three parts. The first part is found under the assumption that the circuit is built up out of a real counter and ideal gates (no delay) in the second and third stage. The "counter" error is reasoned in

- different rise and fall times of counter flip-flops and tolerances of rise and fall time of counter flip-flops (time lag τ_z),
- propagation delay when using an asynchronous counter (time lag $(n-1)\tau_z$).

The second part arises from the assumption, that counter flip-flops as well as output gates are ideal and that real gates are taken for the logical operations in the second stage. This "logical" error ϵ_L is caused by the different number of gates which triggering signals must pass to produce different Walsh-functions (time lag τ_L) and the difference of rise and fall times and its tolerances of these gates (time lag $\Delta\tau_L$).

At last the third part arises from the assumption of an ideal counter stage, ideal logical operations stage and real output gates. This "output" error is reasoned in the differing rise and fall times and its tolerances of the output gates (time lag $\Delta\tau_G$).

References.

- [1] Harmuth, H., "A generalized concept of frequency and some applications", IEEE Trans. on Information Theory Vol. IT-14, pp. 375-382, May 1968.
- [2] Harmuth, H., "Transmission of information by orthogonal functions", Springer, New York, 1969
- [3] Lebert, F.J., "Walsh function generator for a million different functions", Proceedings of symposium on applications of Walsh functions, Naval Research.

Laboratory, Washington DC, USA, 1970,
pp. 58-61.

[4] Yuen, C.K., "New Walsh-function
generator", Electronics Letters Vol. 7,
pp. 605-607, Oct. 1971.

AN ORTHOGONAL TRANSFORM APPROACH TO THE DESCRIPTION OF BIOLOGICAL AND MEDICAL SYSTEMS

Fritz J. Seif and Donald S. Gann

Medizinische Universitäts-Poliklinik, D7400 Tübingen,
West Germany, and Department of Biomedical Engineering,
Johns Hopkins University, School of Medicine, Baltimore,
Maryland, USA.

Abstract

In biology and medicine a systems description in discrete form is mandatory and can be based on a Boolean algebra maintaining a quantitative and deterministic relationship between the variables. In contrast to previous approaches (GANN, SCHOEFFLER), the mapping of the logic systems description into a set of orthogonal functions over a Galois field leads to computational advantages. The applicability of the mathematical technique to biological systems description is demonstrated by the human female reproductive cycle as an example.

Introduction

In biology and medicine systems description serves the purpose of summarizing our knowledge of a system. A mathematical representation can help to check a complex hypothesis or to predict results by simulation. Therefore (empirical) data and (theoretical) assumptions must be incorporated in a mathematical model.

Empirical information of a system can only be obtained by observables and be represented by a sequence of discriminatory states and a diagram of information flow together with the memory depth of the system. This representation of data can be extended by assumptions to meet a certain hypothesis or to fill a gap of knowledge.

As we want a definite answer, decision, or feedback by a simulation, only deterministic systems are of interest. By Heisenberg's principle of indeterminacy measurements yield only discrete data of finite resolution, if we exclude any probabilistic statements. For the same reason only disjoint equivalence sets of state variables $x_i(t)$, $i = 1, 2, 3, \dots, n$ and state sequences $\hat{x} = 1, 2, 3, \dots$ are allowed. Thus the model of the system is conceived as a deterministic finite automaton or sequential machine.

Supported by a fellowship of the Deutsche Forschungsgemeinschaft (F.J.S.) and by NIH Grant AM 14952 (D.S.G.).

Mathematics

A deterministic systems description can be obtained by assigning discrete variables, i.e. two-valued Boolean functions $x_i(t)$ to the observables (GANN, SCHOEFFLER). To each qualitative observable - with the first order of resolution - only one Boolean function is attributed, whereas quantitative observables $x(t)$ are quantized by the dyadic expansion into several two-valued functions $x_i(t)$:

$$x(t) \leq \sum_{i=1}^n 2^{n-i} x_i(t). \quad (1)$$

$t=1, 2, 3, \dots, 2^n$ is the attributed state out of the state sequence.

A complete state description in the discrete variables $x_i(t)$, $i=1, 2, \dots, n$ at the state t is obtained by an elementary Boolean conjunction

$$y(t) = \prod_{i=1}^n x_i'(t) \quad (2)$$

with $i=1, 2, \dots, n$; $x_i' = x_i$ or \bar{x}_i and with the operators "AND" (usually written as concatenation) and \bar{x} as "NEGATION" of x .

Thus in a system with n variables maximally 2^n different states can be observed forming the same number of elementary conjunctions. The system is called to be of n -th order.

The dynamic behavior of the system can be represented by a consecution of states

$$\begin{array}{c|c|c|c|c|c} t & 1 & 2 & \dots & t & t+1 \dots \\ \hline y(t) & \prod_{i=1}^n x_i'(1) & \prod_{i=1}^n x_i'(2) & \dots & \prod_{i=1}^n x_i'(t) & \dots \end{array}$$

or by a state transition matrix, as usually applied in the description of Markovian processes. The state transition matrix T (in general $2^n \times 2^n$) maps all possible state descriptions $y(t-1)$ at state $t-1$, represented by the vector $y(t-1)$, into all possible state description $y(t)$ at the next state t , represented by $y(t)$:

$$y_n(t) = T_n y_n(t-1). \quad (3)$$

In other words T_n maps the vector space Y of y_n into itself.

We call $y_n(t)$ an intensional vector, and want to distinguish it from another vector to be introduced later as an extensional or spectral vector (LEIM-FELLNER).

The intensional vector $y_n(t)$ is called to be of n -th order with n representing the number of discrete variables of the system. The elements of the vector $y_n(t)$ amount to 2^n and can be obtained by the ordered Cartesian product of all n two-valued elementary vectors. Omitting the state index t , we write:

$$y_n = \begin{pmatrix} x_1 \\ \bar{x}_1 \end{pmatrix} \times \begin{pmatrix} x_2 \\ \bar{x}_2 \end{pmatrix} \times \dots \times \begin{pmatrix} x_n \\ \bar{x}_n \end{pmatrix} = y_{n-1} \times \begin{pmatrix} x_n \\ \bar{x}_n \end{pmatrix} \quad (4)$$

For example $n=3$ yields:

$$y_3 = (x_1 x_2 x_3, x_1 x_2 \bar{x}_3, x_1 \bar{x}_2 x_3, x_1 \bar{x}_2 \bar{x}_3, \bar{x}_1 x_2 x_3, \bar{x}_1 x_2 \bar{x}_3, \bar{x}_1 \bar{x}_2 x_3, \bar{x}_1 \bar{x}_2 \bar{x}_3).$$

Equation (3) can be considered as an overall transfer function of the system. In order to obtain more insight into the structure of the system and the ways of information flow, it would be desirable to have an explicit solution for the single discrete variable $x_i(t)$ in the form:

$$x_i(t) = f(x_1(t-1), x_2(t-1), \dots, x_n(t-1)) \quad (5)$$

This can be achieved by three different basis transformations in (3). In the intensional vector y_n the variables of the elementary conjunctions x_i are substituted (DE LUCA, HAMMER) by the following:

H-transformation:

$$x_i = 1/2(1+r_i) \quad \wedge = . \\ \bar{x}_i = 1/2(1-r_i) \quad \vee = +$$

or in vectorial form

$$y_1 = \begin{pmatrix} x_1 \\ \bar{x}_1 \end{pmatrix} = 1/2 \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{pmatrix} 1 \\ r_1 \end{pmatrix}; \\ y_1 = 1/2 H_1 E_1 = H_1^T E_1 \quad (6)$$

H_1 : elementary transformation matrix

H_1 : elementary Hadamard matrix

E_1 : elementary spectral Rademacher-Walsh vector

r_i : Rademacher function of i -th order.

P-transformation:

$$x_i = u_i, \bar{x}_i = 1-u_i, \wedge = .; \vee = +$$

or in vectorial form

$$y_1 = \begin{pmatrix} x_1 \\ \bar{x}_1 \end{pmatrix} = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{pmatrix} u_1 \\ 1 \end{pmatrix}$$

$$y_1 = P_1 u_1 \quad (7)$$

P_1 : elementary transformation matrix

u_1 : elementary spectral vector

S-transformation:

$$x_i = x_i \quad \wedge = . \\ \bar{x}_i = 1 \oplus x_i \quad \vee = \oplus$$

\oplus denotes the logic exclusive OR.

$$y_1 = \begin{pmatrix} x_1 \\ \bar{x}_1 \end{pmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{pmatrix} x_1 \\ 1 \end{pmatrix}$$

$$y_1 = S_1 x_1 \quad (8)$$

S_1 : elementary transformation matrix

x_1 : elementary extensional Boolean vector.

The H- and P-transformation maps from a Boolean space into the real space, whereas the S-transformation projects into another Boolean space, which is isomorphic to a Galois field mod 2 GF(2).

The elements of the transformation matrices of the n -th order are obtained by the n -times Kronecker product (\otimes) of the elementary matrices, as the latter are given over a finite field (GRÖBNER):

$$H_n' = H_1' \otimes H_1' \otimes \dots \otimes H_1' \quad (n\text{-times}) \\ = H_{n-1}' \otimes H_1' \quad (9)$$

or in the form of the normalized Hadamard matrix

$$1/2^n H_n = 1/2^{n-1} H_{n-1} \otimes 1/2 H_1 \quad (10)$$

The same holds for P_n and S_n :

$$P_n = P_{n-1} \otimes P_1 \quad (11)$$

$$S_n = S_{n-1} \otimes S_1 \quad (12)$$

In the case of H_1, H_1 and P_1 the operations $+$, $-$, and \cdot are used, whereas in the case of S_1 the operations \oplus and \wedge in the finite field GF(2) with addition modulo 2, with $1 \oplus 1 = 0$, $1 \oplus 0 = 0 \oplus 1 = 1$, and $0 \oplus 0 = 0$.

It is easily shown that the elementary matrices have the following properties:

H_1 is orthogonal under the operations $+$ and \cdot .

$$1/2^n H_n H_n = E_n \quad (13)$$

with $H_n^{-1} = H_n^T = H_n$, and E_n is the identity matrix.

P_1 is not orthogonal, with $P_1 \neq P_1^{-1}$

and $P_1^{-1} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ we have

$$P_1^{-1} P_1 = P_1 P_1^{-1} = E_1 \quad (14)$$

S_1 is orthogonal under the operations \circ and \wedge .

With $S_1^{-1} = S_1$ and $(S_1^T)^{-1} = S_1^T$ we have

$$S_1 S_1 = E_1 \quad (15)$$

As the Kronecker product preserves the properties of the elementary matrices (GRÖBNER), we have:

$$1/2^n H_n H_n = E_n \quad (16)$$

$$P_n^{-1} P_n = P_n P_n^{-1} = E_n \quad (17)$$

$$S_n S_n = E_n \quad (18)$$

The elements of the vectors \underline{r}_n , \underline{u}_n , and \underline{x}_n of n -th order we obtain also by the Cartesian product: For example

$$\underline{r}_n = \underline{r}_{n-1} \times \begin{pmatrix} 1 \\ r_n \end{pmatrix} \text{ and}$$

$$\underline{x}_n = \underline{x}_{n-1} \times \begin{pmatrix} x_n \\ 1 \end{pmatrix},$$

with $n = 3$ we get

$$\underline{r}_3 = (1, r_3, r_2, r_2 r_3, r_1, r_1 r_3, r_1 r_2, r_1 r_2 r_3)$$

$$\underline{x}_3 = (x_1 x_2 x_3, x_1 x_2, x_1 x_3, x_1, x_2 x_3, x_2, x_3, 1)$$

The elements of the spectral vector \underline{r}_n represent the Walsh functions in the natural Walsh-Paley order written as products of Rademacher functions. Here we replace the nonlinear Rademacher products $r_1 r_2 r_3 \dots r_n$ by auxiliary variables, the Walsh functions, in the sense of a Hamilton transformation. This "linearization" together with the intrinsic properties of the Rademacher functions leads to the advantage of solving logic (Boolean) equations by matrix operations despite the underlying nonlinearities. The same is true with \underline{u}_n and \underline{x}_n .

For the intensional vector \underline{y}_n of n -th order we thus have the three basis transformations:

$$\underline{y}_n = 1/2^n H_n \underline{r}_n \quad (19)$$

$$\underline{y}_n = P_n \underline{u}_n \quad (20)$$

$$\underline{y}_n = S_n \underline{x}_n \quad (21)$$

The substitution of (19), (20), and (21) into (3) yields:

$$1/2^n H_n \underline{r}_n(t) = 1/2^n T_n H_n \underline{r}_n(t-1)$$

and with (16)

$$\begin{aligned} \underline{r}_n(t) &= 1/2^n H_n T_n H_n \underline{r}_n(t-1) \\ &= 1/2^n A_n \underline{r}_n(t-1) \end{aligned} \quad (22)$$

$$A_n = H_n T_n H_n \text{ and } T_n = 1/2^{2n} H_n A_n H_n; \quad (23)$$

$$P_n \underline{u}_n(t) = T_n P_n \underline{u}_n(t-1)$$

and with (17)

$$\underline{u}_n(t) = P_n^{-1} T_n P_n \underline{u}_n(t-1)$$

$$\underline{u}_n(t) = B_n \underline{u}_n(t-1) \quad (24)$$

$$B_n = P_n^{-1} T_n P_n \text{ and } T_n = P_n B_n P_n^{-1}; \quad (25)$$

$$S_n \underline{x}_n(t) = T_n S_n \underline{x}_n(t-1)$$

and with (18)

$$\begin{aligned} \underline{x}_n(t) &= S_n T_n S_n \underline{x}_n(t-1) \\ &= C_n \underline{x}_n(t-1) \end{aligned} \quad (26)$$

$$C_n = S_n T_n S_n \text{ and } T_n = S_n C_n S_n. \quad (27)$$

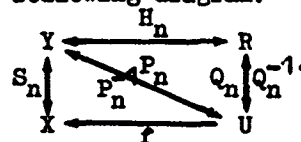
Thus we obtain among others the explicit solutions for

$$r_1(t), r_2(t), \dots, r_n(t);$$

$$u_1(t), u_2(t), \dots, u_n(t);$$

$$x_1(t), x_2(t), \dots, x_n(t);$$

which represent the structural dependency of one variable at t from the variables at the former state $t-1$. How to reach the explicit solution in X , given a solution in R or U , is depicted in the following diagram:



with $\underline{u}_n = 1/2^n P_n^{-1} H_n \underline{r}_n$, $1/2^n Q_n \underline{r}_n$ and

$$Q_1 = P_1^{-1} H_1 = \begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix}$$

and $f: \pm = \circ$.

The structural dependencies in X tell us a possible way of information flow in the system.

Equations (23), (25), and (27) show how to solve the inverse problem of finding the state transitions given the structural relationship between the variables. A_n , B_n , and C_n we call the structure matrix of order n of the system.

So far we have silently assumed that the state t is completely determined by the previous state $t-1$. In reality this is seldom true; the state t might as well depend on earlier states $t-j$, $j=1, 2, 3, \dots, 2^n-1$, representing the memory depth of the system. Under these circumstances we write:

$$\begin{aligned} \underline{y}_n(t) &= T_{n,h}(\underline{y}_n(t-1) \times \underline{y}_n(t-2) \times \dots \\ &\quad \dots \times \underline{y}_n(t-j) \dots) \\ &= T_{n,h} \underline{y}_h(t') \end{aligned} \quad (28)$$

with h representing the order of the newly formed vector of the "previous" states. The order h can increase to $n(2^n-1)$ maximally. The number of the elements of $\underline{y}_h(t')$ and the rows of $T_{n,h}$ will then be $2^n = 2^{n(2^n-1)}$, and the

number of the columns 2^n . The structural relationship between the state variables will be obtained, for example by the S-transform, by substituting (21) into (28) and we have:

$$\underline{x}_n(t) = S_n T_{n,h} S_n \underline{x}_n(t') \quad (29)$$

Static systems can be described by the structural relationship

$$\underline{x}_n = C_n \underline{x}_n \quad (30)$$

in the X domain.

For a system with memory depth of one state it can easily be shown that the following holds:

$$\underline{y}_n(t) = T_n^j \underline{y}_n(t-j) \quad (31)$$

with $j = 1, 2, 3, \dots, 2^n - 1$.

If for some $j=e$, $\underline{y}_n(t) = \underline{y}_n(t-e)$, or $T_n^e = E_n$ with the rank r of T_n and E_n $r=2^n$, we call T_n cyclic. It has got the properties of a permutation matrix and (3) can be inverted to (OULL):

$$\underline{y}_n(t-1) = T_n^{-1} \underline{y}_n(t) \quad (32)$$

With these properties the system under consideration shows cyclic or periodic behavior with a cycle length or period of e .

In case of

$$T_n^1, T_n^2, T_n^3, \dots, T_n^e = T_n^{e+1} = T_n^{e+2} = \dots$$

we call T_n transient with the characteristic exponent e . The system has got a transient length of e , i.e. starting in state t the system reaches the limiting state $t+e$ or the state of equifinality (VON BERTALANFFY) after e transitions. We can write:

$$\underline{y}_n(t+e) = T_n^e \underline{y}_n(t),$$

or generally

$$\underline{y}_n(t+e+a) = T_n^{e+a} \underline{y}_n(t) = T_n^e \underline{y}_n(t) \quad (33)$$

with $a=0, 1, 2, 3, \dots, \infty$.

In the limiting state $t+e$ the system has lost all information of any previous state. This gives some plausibility, why a transient matrix is not invertible.

There also exist systems and thus matrices that are cyclic and transient. The cyclic and transient properties of the state transition matrix T_n are also inherent in the structure matrices A_n , B_n , and C_n , as the H-, P-, and S-transformations preserve these properties (GRÜBNER).

Application

As an example out of the field of biology and medicine we have chosen the human female reproductive cycle or menstrual cycle. It evolves in time with a period of about 28 days. In

simplifying we can say that the pituitary gland, anatomically and functionally connected to the hypothalamus, produces two hormones as information carriers, the follicle stimulating hormone (FSH) and the luteinizing hormone (LH), which stimulate the ovaries with respect to follicle growth, maturation, ovulation, and steroid (sex) hormone production.

The ovarian steroid hormones comprise two groups of different biological activities, the estrogens and gestagens. Both control the LH- and FSH-production and secretion in the pituitary by a negative feedback. In a certain state of the system the estrogens elicit also a positive feedback on the pituitary, which causes the FSH- and LH-surge in the middle of the cycle and is followed by the ovulation and a considerable increase of the gestagen production by the ovary during the second half of the cycle. Estrogens and gestagens also induce proliferation and secretion in the endometrium. At the end of the cycle the hormonal decline causes the menstrual bleeding.

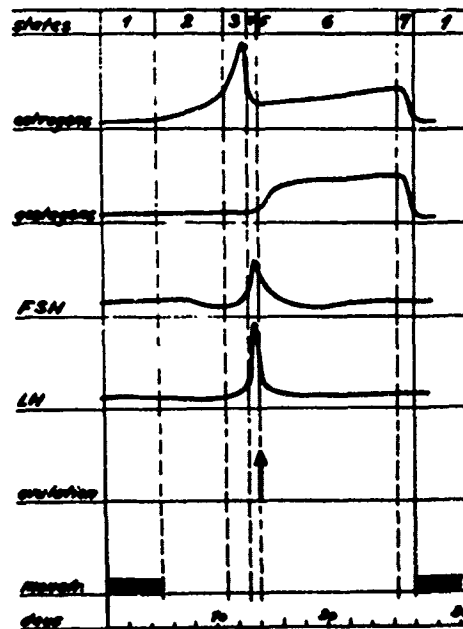


Fig. 1

The discrete events in the system and the time dependencies of the hormone levels in the blood are depicted in Fig.1. Further simplification and abstraction to discriminatory states yield Fig.2a and Fig.3a. Seven different states ($t=1, 2, \dots, 7$) can be distinguished and described by 8 two-valued variables x_1, x_2, \dots, x_8 . The so far known structure of the information flow in the system can be seen in Fig.4.

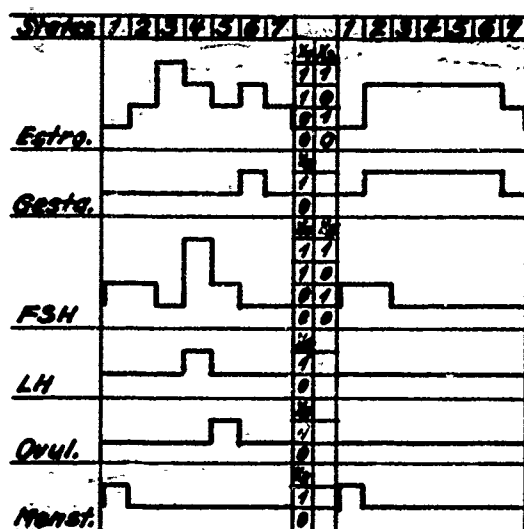


Fig. 2 a b

Tabulation of the dyadic state variables

Normal cycle								Simulation							
1	2	3	4	5	6	7	State	1	2	3	4	5	6	7	
0	0	1	1	0	1	0	x_1	0	1	1	1	1	1	0	
0	1	1	0	1	0	1	x_2	0	0	0	0	0	0	1	
0	0	0	0	0	1	0	x_3	0	1	1	1	1	1	0	
0	0	0	1	0	0	0	x_4	0	0	0	0	0	0	0	
1	1	0	1	1	0	0	x_5	1	1	0	0	0	0	0	
0	0	0	1	0	0	0	x_6	0	0	0	0	0	0	0	
0	0	0	0	1	0	0	x_7	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	x_8	1	0	0	0	0	0	0	

Fig. 3 a b

The problem to be solved is to find a transfer function of the "black boxes" G_1, G_2, \dots, G_8 , representing subsystems, which generate $x_i(t)$, $i=1, 2, \dots, 8$ out of $x_i(t-1)$. In general the memory depth is assumed to be one state, except for x_6, x_7 , and x_8 , in which case a memory depth of 2 states is necessary to obtain determinancy.

With the mathematical technique displayed in the former chapter we obtain the explicit solutions or transfer functions in the Rademacher-Walsh domain by the Hadamard (H-) transform:

With $r_1(t)=r_1$, $r_1(t-1)=r_1'$, $r_1(t-2)=r_1''$ we get

$$r_1 = 1/2^6 (1+r_2')(1-r_3')(1-r_4')(1-r_6') \cdot (2(1-r_1')(1+r_5')+(1+r_1')(1-r_5')(1-r_7'))-1$$

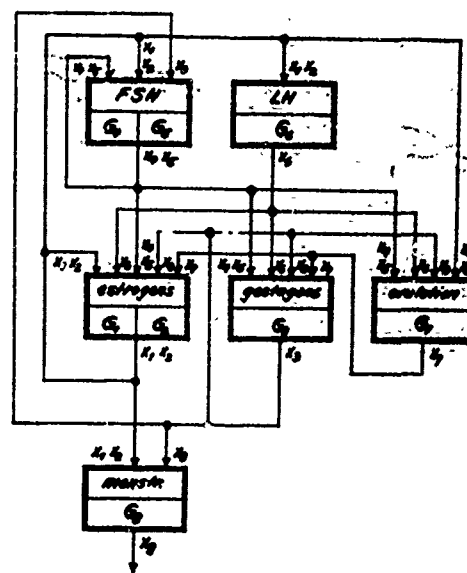


Fig. 4

$$r_2 = 1/2^6 (1-r_7')(2(1-r_1')(1-r_3')(1-r_4') \cdot (1+r_5')(1-r_6')+(1+r_1')(1-r_2')((1-r_3') \cdot (1+r_4')(1+r_5')(1+r_6')+(1+r_3')(1-r_4') \cdot (1-r_5')(1-r_6')))-1$$

$$r_3 = 1/2^4 (1-r_3')(1-r_4')(1+r_5')(1-r_6')(1+r_7')-1$$

$$r_4 = 1/2^2 (1+r_1')(1+r_2')(1-r_3')-1$$

$$r_5 = 1/2 (1-r_3')(1-r_2'r_5')-1$$

$$r_6 = 1/2^3 (1+r_1')(1+r_2')(1-r_4')(1+r_2'')-1$$

$$r_7 = 1/2^8 (1+r_1')(1-r_2')(1-r_3')(1+r_4')(1+r_5') \cdot (1+r_6')(1-r_4'')(1-r_5'')(1-r_6'')-1$$

$$r_8 = 1/2^5 (1-r_1')(1+r_2')(1-r_3')(1+r_4') \cdot (1-r_2'')(1+r_3'')-1$$

And in the logic domain by the S-transform with $x_i(t)=x_i$, $x_i(t-1)=x_i'$, $x_i(t-2)=x_i''$ we have:

$$x_1 = x_2'(1 \oplus x_3')(1 \oplus x_4')(1 \oplus x_6')(x_1' \oplus x_5'(1 \oplus x_1'(1 \oplus x_7'))))$$

$$x_2 = (1 \oplus x_7')((1 \oplus x_1')(1 \oplus x_3')(1 \oplus x_4')x_5'(1 \oplus x_6') \cdot x_1'(1 \oplus x_2')((1 \oplus x_3')x_4'x_5'x_6' \cdot x_3'(1 \oplus x_4')(1 \oplus x_5')(1 \oplus x_6'))))$$

$$x_3 = x_5'x_7'(1 \oplus x_3')(1 \oplus x_4')(1 \oplus x_6')$$

$$x_4 = x_1'x_2' \oplus x_1'x_2'x_3'$$

$$x_5 = x_2' \oplus x_5' \oplus x_2'x_3' \oplus x_3'x_5'$$

$$x_6 = x_1'x_2'x_2''(1 \oplus x_1'')$$

$$x_7 = x_1^1 x_4^1 x_5^1 x_6^1 (1x_2^1) (1x_3^1) (1x_4^1) \\ (1x_5^1) (1x_6^1)$$

$$x_8 = x_2^1 x_1^1 x_3^1 (1x_4^1) (1x_5^1) (1x_6^1)$$

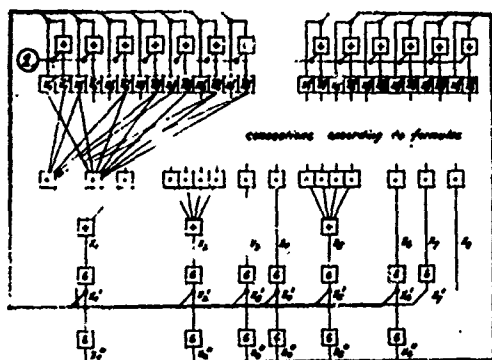


Fig. 5

According to the above formulas circuits can be constructed that behave analogously to the quantized observables of the menstrual cycle. A possible realization of the network in the logic S-domain, with a DC source of unity (1), delays (*), and the logic elements "exclusive OR" (+) and "AND" (.), is shown in Fig. 5.

The effects of a contraceptive estrogen and gestagen medication, as shown in the two upper curves in Fig. 2b and the three upper rows in Fig. 3b can be simulated. The results are seen in the lower parts of Fig. 2b and 3b and are consistent with the biological findings of a missing FSH- and LH-surge and the omission of the ovulation.

Conclusion

The application of this mathematical technique is not limited to the female reproductive cycle, but can be applied to other biological and medical systems for the purpose of description, simulation, and prediction. It can equally well be used for the design of logic switching circuits.

By the H-, P-, and S-transform the intensional state description of the system in the Boolean domain is projected into other domains, where matrix operations are possible and explicit solutions can be obtained for the discrete logic variables. The computational simplicity is greatest in the logic exclusive-OR domain X, lower in the Rademacher-Walsh domain R, and lowest in U.

In our example of a systems description we have incorporated only pertinent information in a discrete form. Thereby we have circumvented the unsurmountable difficulties of a continuous description in the realm of biology and excluded the pseudoaccuracies resulting from differential and integral calculus.

References

1. L. von Bertalanffy: General Systems Theory, G. Braziller, New York 1968.
2. P. Cull: Linear Analysis in Switching Nets, Kybernetik, 8(1971).
3. D.S. Gann, L.E. Ostrander and Schoeffler: A Finite State Model of the Control of Adrenal Cortical Steroid Secretion, in M.D. Mesarović, Systems Theory and Biology, Springer-Verlag, Berlin-Heidelberg-New York 1968.
4. W. Gröbner: Matrizenrechnung, Bibliographisches Institut, Mannheim 1966.
5. P.L. Hammer (Ivanescu) and S. Rudeanu: Boolean Methods in Operation Research and Related Areas, Springer-Verlag, New York 1968.
6. W. Leinfellner: Einführung in die Erkenntnis- und Wissenschaftstheorie, Bibliographisches Institut, Mannheim 1965.
7. A. de Luca: On some Representations of Boolean Functions. Application to the Theory of Switching Elements Nets, Kybernetik, 9(1971), 1 - 10.
8. J.D. Schoeffler, L.E. Ostrander and D.S. Gann: Identification of Boolean Mathematical Models, as in l.c. 3..

A QUANTIZED VARIABLE APPROACH TO DESCRIPTION OF BIOLOGICAL AND MEDICAL SYSTEMS

Donald J. Gann, Fritz J. Seif and James D. Schoeffler
Department of Biomedical Engineering, The Johns Hopkins
University School of Medicine, Baltimore, Maryland and
Systems Research Center, Case Western Reserve University,
Cleveland, Ohio.

Abstract

Models of biological systems are often based on sampled data in which the values of variables are associated with large variances. Under these conditions it is traditional to assume continuity in time and amplitude, so that time is continuous and each amplitude is an element of the real line. An alternative approach is to utilize a discrete time scale with time identified by points at which samples are obtained (sampled data approach) or at which changes of state occur (discrete event approach). Furthermore, variables may be quantized to distinguish only that number of levels of a variable which may be distinguished statistically. This results in an ordinal scale of measurement. In our experience it is rare to require more than eight levels of an experimental variable, using the above criterion. The mathematical handling of such discrete variables may be greatly simplified by coding each variable into a Boolean vector. This approach also eliminates certain problems of nonlinearity. State variables which have not been observed are distinguished in such a representation as unknown parameters which are themselves Boolean elements. These parameters can be identified by the solution of simultaneous Boolean equations. These equations arise by application of a constraint of functionality (determinism) whereby a given input applied at a given state must lead to a unique output. A technique for solving such equations has been evolved for the Boolean algebra of n -dimensional Boolean vectors, conjunction and inclusive disjunction. Simplification following parameter identification leads to compact representations. The finite cardinality of the models aids in identification of inconsistencies and thus of critical experiments, and aids in the consideration of the relative validity of alternative structures. A simple theorem relates the disjunctive normal representation of a Boolean sequence, in which the elements are regarded as integers, to the Walsh Transform, and thus relates a logical formulation to an (arithmetic) algebraic one. Boolean minterms can then be identified by solution of simultaneous algebraic equations. The relation is of potential significance for applications to large systems of the variety described above.

Introduction

Models of biological systems are formulated to summarize complicated sets of data, to serve as a basis for simulation and thus for prediction of the outcomes as-yet-unperformed experiments, or to increase one's understanding of the real system described by the model. The process of modeling inevitably begins with the abstraction

tion of the real system in order to associate certain mathematical objects with elements of the real system. Inevitably, certain features of the real system are emphasized at the expense of others. The best models are those which are isomorphic to the real system with respect to those features which are considered critical and which have motivated the construction of the model in the first place. Several properties of biological systems, and certain aspects of biology as a science, suggest the validity of constructing models of biological or medical systems which utilize variables quantized in both amplitude and time.

Of primary importance, the major secondary problems in such areas as physiological control systems and medical diagnosis tend to emphasize aspects of the connectivity of systems. The problem may be stated, in a complex system in which all the elements are interconnected, which connectors are of critical importance? What is the relative importance of the critical connections? If the system is malfunctioning, is a connection disturbed, or has the relative importance of the connections changed? Alongside these questions, the need for precise measurement may be minimized.

Furthermore, biological systems tend to be noisy, and measurements on them often are associated with large variances. As Chang (1) has pointed out, there is a natural quantization associated with the signal/noise ratio, and with all band-limited signals. As variances become high, the number of levels into which a signal can be resolved becomes quite small. Li and Urquhart (2) have analyzed the secretory properties of the adrenal gland in this way, and have shown that it can be viewed as secretory not continuously, but at only eight different rates. In addition to large variances, certain measurements can be made only at discrete intervals. This is always the case for samples of body fluids or tissues which must be submitted to biochemical analysis: it is also the case for so-called continuous measurements in which the time constant of the measurement is large with respect to the dynamic properties of the system. The measurement of cardiac output, which is based on the dilution of an injected dye, is one example: since the phenomenon of distribution must be measured over a period of many heartbeats. Thus quantization may be implicit either in the system or in the process of measurement.

The present paper describes a technique for

the modeling of systems with variables quantized in amplitude and time which extends previous descriptions of this approach (3,4,5). It is shown that the problem of identification of parameters in such a model reduces to the problem of the solution of simultaneous Boolean equations, and a new method is introduced to accomplish this. A relation of this formulation to the Walsh transform is examined. An extension of the present approach which was suggested by this examination, but which offers certain computational advantages, is described in an accompanying paper (6).

The Formulation of Boolean Models

Given a system to be modeled, it is first necessary to abstract the system, to define the sets of inputs and outputs and to formulate an initial view of the connectivity of the system. The variables can be quantized in amplitude according to Chang's method (1), or in any other intuitively useful way. For example, the process of measurement or the design of experiments may suggest appropriate approaches to quantization (3,4). Quantization in time may be based on sampling intervals (sampled data approach, described below) or on times at which changes in state occur (discrete event approach (See (7))). The differences between levels of the amplitude variables need not be equal: only linear ordering is required. The scale of measurement for amplitude is thus ordinal. The amplitude variables are then coded into appropriate Boolean vectors. Any code will suffice. We have in general used the natural equivalent to a binary representation of the ordinal number. On the other hand, Brand (18) has shown that the Gray code has certain advantages if there are ambiguities in the quantization. Since the mapping from ordinal measure to Boolean vector is one-to-one, the decoding procedure is implicit. In general the model will take the format of a set of interconnected subsystems. As a result of the quantization, there can be only a finite number of inputs to the system or to any of the subsystems. One might be able, if all intermediate variables are known, to write a truth table for each subsystem and for the system as a whole which will list all possible outputs for all possible inputs and times (4). In general, there will be one or more intermediate variables which cannot be observed. This gives rise to the problem of identification: to find the values of all unobserved variables. This procedure would be entirely arbitrary if we did not want to be able to use the model for prediction. However, predictability implies determinism, which introduces in turn the constraint of functionality: for a given input and state, there can be one and only one output. This constraint, applied to each subsystem implies that if in any two rows of a truth table the inputs are equal, then for these rows, the outputs must also be equal (4). Thus for a subsystem with input r_i and output s_j one could write

$$(r_i = r_j) \longrightarrow (s_i = s_j), \quad (1)$$

where \longrightarrow stands for implication

from the axioms of Boolean algebra, equation (1)

becomes

$$(r_1 r_2 + \bar{r}_1 \bar{r}_2) (s_1 s_2 + \bar{s}_1 \bar{s}_2) \quad (2)$$

Since the statement $a \equiv b$ is equivalent to $a+b$, equation (2) can be written in matrix form as

$$\begin{bmatrix} 0010 & 1101 \\ 0001 & 1110 \\ 1110 & 0001 \\ 1101 & 0010 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (3)$$

In general, for a subsystem with multiple inputs and outputs, the identification problem will then reduce to the problem of solving an equation of the form

$$AX + B\bar{X} = \underline{1}, \quad (4)$$

where X is an n -vector of the n inputs and outputs, A and B are Boolean $m \times n$ matrices and $\underline{1}$ is an m -vector all elements of which are 1. A method for the solution of this class of equations is given below.

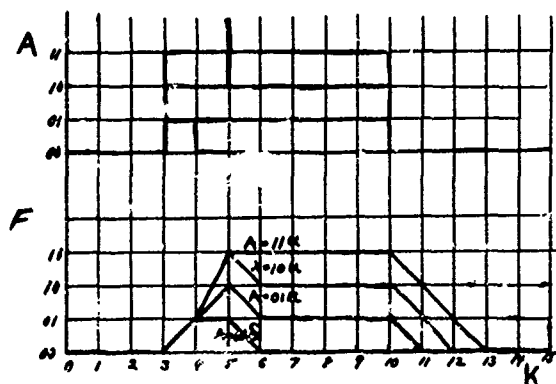
Examples of static (3) and discrete event (7) Boolean models have been presented elsewhere. The static example is particularly interesting in light of the discussion above; since it was shown that the initial structure (connectivity) chosen for the system did not allow any solution to the simultaneous equations for one of the subsystems. Accordingly an alternate connectivity was selected which appeared compatible with the experimental evidence. Further experiments were conducted to test the new connectivity, and this was shown to be correct (9). Thus, application of this modeling procedure led to a novel, hitherto unsuspected and important physiological finding concerning a hormonal action on the central nervous system. The discrete-event example is interesting because it offers a Boolean model of a technological process (in this case a conveyor system) and thus indicates the applicability of this approach to non-biological cases. Aveyard (7) showed that the discrete event approach was dimensionally smaller and thus simpler than an equivalent model of the system as an automaton. Nevertheless, there are occasions in which one may prefer the automata approach. One such circumstance is that in which inputs and outputs are sampled periodically, and one seeks a model to relate them in the absence of knowledge of the mechanisms relating inputs and outputs. An example follows.

A Sampled-Data Dynamic Boolean Model

The adrenal gland of many mammals secretes a hormone, cortisol, in response to the presentation of a hormone from the pituitary gland, corticotrophin, or ACTH. ACTH is released by the pituitary in response to a variety of stimuli generally termed "stress," and cortisol plays a central role in the defenses of the body against hemorrhage, injury and other stresses (10). For reasons given elsewhere (3) it seems desirable to model the overall system in Boolean form. Urquhart and Li (11) have described in detail the dynamics of the adrenal response to

ACTH, and have offered a continuous model to account for these dynamics. The adrenal displays a static nonlinearity so that its response saturates at large inputs of ACTH. There is also a dynamic asymmetry such that an overshoot is present in the response to presentation of ACTH (at submaximal levels only), but there is no undershoot in response to withdrawal of ACTH. All of these properties can be displayed if cortisol secretion is viewed in four levels (2 bits), though the gland itself actually behaves as if it operates in eight levels (3 bits) (2). We offer here this simplified model of the adrenal response to ACTH, based on a 2-bit processor, as an illustration of a sampled-data dynamic Boolean model.

The (sampled) output of secretion of cortisol in response to presentation of ACTH as a unit pulse and as pulses of three different amplitudes and longer duration is shown, after quantization, in Figure 1. The overshoot and saturation are evident. For convenience the differences between levels of the output variable F have been drawn as equal. In fact the distance from $F = 10$ to $F = 11$ should be shorter than that from $F = 01$ to $F = 10$.



A truth table was constructed to relate a state vector Z to the two past outputs, each of which is a 2-vector, to the next output, and to the past input. The constraint of functionality was applied to define the state elements, as indicated above. For a six-component state vector Z , for the output at time k , $f_k = (f_1, f_2)$ and the output at time $k-1$, $f_{k-1} = (f_3, f_4)$ and for the input at time k , $a_k = (a_1, a_2)$, the state equations are simply:

$$\begin{aligned} z_1 &= f_2 \bar{f}_3 \bar{f}_4 \\ z_2 &= f_1 + f_2 & z_5 &= a_1 \\ z_3 &= f_1 & z_6 &= a_2 \\ z_4 &= f_2 \end{aligned}$$

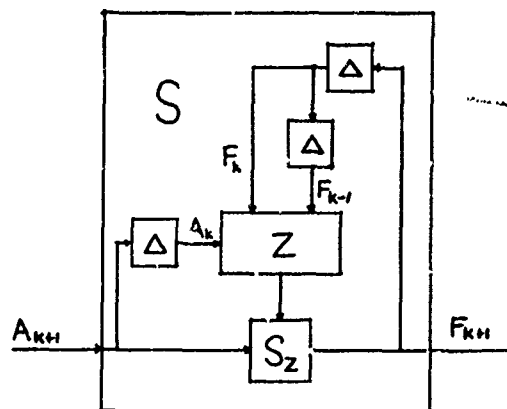
A truth table was next constructed for the overall system relating the inputs and present state to the present output. The missing elements of the truth table were derived from the

identification procedure outlined above. The overall dynamic model for the system as two Boolean equations is relatively compact, yet describes in full the complexity of the data of Figure 1. Furthermore, the model is complete: simulation will give an output for any waveform of input, restricted to frequencies at or below the sampling rate. The overall equations are for output at time

$$k+1, f_{k+1} = (f_1', f_2'):$$

$$\begin{aligned} f_1' &= \bar{z}_1 z_2 z_3 z_4 \bar{z}_5 + \bar{z}_1 z_2 z_5 (z_3 + z_4) \\ &\quad + z_1 \bar{z}_2 \bar{z}_3 z_4 (z_5 + z_6) \\ f_1/2 &= \bar{z}_1 z_2 z_3 \bar{z}_4 (z_5 + z_6) + z_1 \bar{z}_2 \bar{z}_3 z_4 \\ &\quad (z_5 + \bar{z}_5 z_6) + \bar{z}_1 z_2 (z_3 \bar{z}_4 \bar{z}_5 + \\ &\quad \bar{z}_3 z_4 z_6 + z_3 z_5 z_6) \end{aligned}$$

The overall model of the adrenal can be represented as a finite automation, as in Figure 2. Δ is a delay of one time-unit.



Solution of Simultaneous Boolean Equations

The solution of Boolean equations seems deceptively simple. In fact, many techniques are available and are effective provided that the number of variables is very small. Most techniques depend upon enumeration of the 2^N possible values of N boolean variables and hence are directly or indirectly techniques for solution based on exhaustive search. Such methods are completely ineffective when the number of variables becomes reasonable (greater than 20). The objective of this section is to describe the technique which has been developed and implemented for the solution of Boolean equations which arise in the applications described in the previous sections. Two criteria are evident: first, the form of the boolean equations and the technique used for their solution must be amenable to digital computer implementation; and second, the solution technique must not be equivalent to enumeration of all possible values of the

variables.

The efficient representation of the equations in computer memory leads to the linear-like form shown:

$$A \underline{X} + B \overline{X} = \underline{1} \quad (5)$$

Here, as before, \underline{X} is an N-vector of the N boolean variables, A and B are $M \times N$ matrices all of whose elements are 0 or 1, $\underline{1}$ is an M-vector all of whose elements are 1, and matrix multiplication is implied in the usual sense except that multiplication is replaced by logical "and" and addition by logical inclusive "or". Thus the first of the M equations is simply

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1N}x_N + b_{11}\overline{x}_1 + \dots + b_{1N}\overline{x}_N = 1$$

In the above equation, either a_{ij} or b_{ij} is zero: for otherwise the equation is redundant, since $x_j + \overline{x}_j$ is identically 1. Notice that the storage of such linear-like equations in memory is straightforward, requiring one bit per variable for the storage of a_{ij} and one bit for b_{ij} . It is convenient in the solution to group L coefficients in each computer word (where L is the word length of the computer) to minimize storage and to take advantage of whole word logical operations available in most digital computers. Hence in a 36-bit machine and a set of 50 variables (say), two words are used for the storage of a row of the A-matrix, and another two words for the storage of the corresponding row of the B-matrix with 22 bits unused in each group of two words. The unused storage is less important when the number of variables is large and is more than made up for by the simplicity of logical operations that result.

It is important to note that the form of simultaneous linear-like equations is completely general:

Theorem 1: Any set of boolean equations involving N variables x_1, x_2, \dots, x_N may be expressed in the form of Equation (5). The proof of this theorem follows immediately from the well known result that any set of simultaneous boolean equations can be written in the form $f(x) = 0$ where the function $f()$ involves the x_i 's and their negations. Moreover, any boolean function in N variables can be expanded into the standard min-term form

$$f(x) = \sum_{i=1}^{2^N} c_i m_i(x) = 0$$

where $m_i(x)$ is the i-th minterm and c_i is zero or 1.

In the above expansion, the symbol "+" corresponds to logical inclusive "or". Since the logical sum is zero, each term must be zero with the result that the equation can be written in the form

$$\{m_i = 0 \mid c_i = 1, i = 1, 2, \dots, 2^N\}$$

That is, any c_i which is zero causes the term to be identically zero. All terms for which $c_i = 1$ then require that the corresponding min-term be zero. Noting that each minterm is the logical product of the x_i or their negations, it follows immediately that negating each simultaneous equation produces the logical sum of variables or their negations, which is the desired form. Hence, any arbitrary Boolean equations can be expressed in the linear-like form which proves convenient for automatic solution on a digital computer. As an example, consider the problem

$$f(x) = x_1x_2\overline{x}_3 + x_1\overline{x}_2x_3 + \overline{x}_1x_2x_3 = 0$$

This function becomes

$$x_1x_2\overline{x}_3 = 0$$

$$x_1\overline{x}_2x_3 = 0$$

$$\overline{x}_1x_2x_3 = 0$$

Negating each equation yields

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \underline{x} + \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \overline{\underline{x}} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

which is the form of Eq. (5).

It should not be inferred from the above proof that this is the easiest or most efficient way to generate the linear-like Boolean equations; for in fact expanding in min-terms results in many extra equations, since often terms can be combined and simplified. Nonetheless, this form does lead to the solution algorithm discussed in the next section.

Solution of Simultaneous Boolean Equations by Successive Elimination of Variables

In order to satisfy the second criterion, and to avoid enumeration of the large number of combinations of values of the Boolean variables, a successive elimination scheme is developed. That is, one variable at a time is chosen to be eliminated, resulting in a new set of equations in the identical form but with one fewer variable. If this is carried out until only one variable remains, the solution for that variable is trivial. Back substitution then permits the determination of all other variables. In the case of multiple solutions, it is possible to generate only one solution at a time.

The successive elimination algorithm proceeds as follows.

(step 1) Pick the variable to be eliminated (called x_1 in this algorithm). Let the vector of remaining variables be called Z . Write the equations in the form

$$\underline{a} x_1 + \underline{b} \overline{x}_1 + A_1 Z + B_1 \overline{Z} = \underline{1}$$

Here \underline{a} and \underline{b} are the first columns of the matrices A and B , and A_1 and

B_i are the remaining columns of A and B. This rewriting is simply for notational convenience.

Denote the i -th equation by

$$(step\ 2) \quad a_i x_i + b_i \bar{x}_i + A_i(i)Z + B_i(i)\bar{Z} = 1$$

Each equation for which $a_i = b_i = 0$ does not involve x_i and hence is added to the reduced set and eliminated from the above set. Note that any equation for which $a_i = b_i = 1$ can be eliminated as redundant.

(step 3)

Each of the remaining equations has either a_i or b_i zero but not both.

For each pair of equations which have $a_i = b_j = 0$, form a new equation

$$[A(i) + A(j)]Z + [B(i) + B(j)]\bar{Z} = 1$$

and add it to the reduced set. Note that if there are M equations with M_i having $a_i = 1$ and $(M-M_i)$ with $a_i = 0$, that $M_i(M-M_i)$ equations results. This completes the reduced set, x_i having been eliminated. The reduced set has the same solution for Z as the original set.

(step 4)

Go to step 1 unless 1 variable remains in the reduced set. In this case, go to step 5. If there are no equations in the reduced set, go to step 7.

(step 5)

The reduced set has only one equation which must be of the form

$$a x_j + b \bar{x}_j = 1$$

If $a = b = 0$, no solution exists. If $a = b = 1$, x_j is arbitrary. To generate one solution, pick a value for x_j (0 or 1) and proceed to step 6.

If $a = 1$ and $b = 0$, $x_j = 1$. If $a = 0$ and $b = 1$, $x_j = 0$.

In either of the last two cases, proceed to step 6 after determining x_j .

(step 6)

If all N variables are determined, halt. Otherwise, recover the previous set of equations which involves one more variable than the set used in step 5. Substitute the values of the known variables. The resulting equations have only one known variable. Proceed to step 5.

(step 7)

This step is reached only if the algorithm results in no equations in the reduced set. This implies that all variables which have not been eliminated are arbitrary. In order to generate one solution, pick any value for these and proceed to step 6.

The above algorithm is guaranteed to determine one solution if at least one solution

to the simultaneous equations exists. Notice that it proceeds by generating new sets of equations, each set in the sequence having one less variable until one or more variables can be determined. Then the sets (which must be saved) are recovered in inverse order, the known variables substituted in order to determine the remaining ones. Notice the analogy to ordinary Gauss elimination in the case of linear algebraic equations.

In order to demonstrate that the algorithm is correct, and that no extraneous solutions are generated or legitimate solutions lost, it is necessary merely to demonstrate the correctness of step 3, the other steps being obvious. To this end, group the equations so that all those with just x_i appear first:

$$x_i + s_i = 1 \quad i = 1, 2, \dots, p$$

The remainder involve \bar{x}_i :

$$\bar{x}_j + s_j = 1 \quad j = 1, 2, \dots, q$$

where s_i and s_j are the remainder of the equation.

These can be written in alternate form by negating each equation:

$$\bar{x}_i \bar{s}_i = 0 \quad i = 1, 2, \dots, p$$

$$x_j \bar{s}_j = 0 \quad j = 1, 2, \dots, q$$

Each group may be combined into a scalar equation:

$$\bar{x}_i \sum_{i=1}^p \bar{s}_i = 0$$

$$x_j \sum_{j=1}^q \bar{s}_j = 0$$

Consider the truth table corresponding to the following two scalar equations:

$$x_j \quad A = 0$$

$$\bar{x}_j \quad B = 0$$

A	B	Solution
0	0	1
0	1	1
1	0	1
1	1	0

Clearly a necessary and sufficient condition for a solution for x_j is that $A \cdot B = 0$. Applying this result to the above scalar equations yields

$$\sum_i \bar{s}_i \sum_j \bar{s}_j = 0$$

Combining the summations changes the form to

$$\sum_{ij} \bar{s}_i \bar{s}_j = 0$$

Now negate the entire equation to get the alternate form

$$\prod_{i,j} s_i + s'_j = 1$$

In order for the logical product of quantities to be 1, each must be 1. Hence this reduces to the set of simultaneous equations

$$s_i + s'_j = 1 \quad i=1,2,\dots,p, \quad j=1,2,\dots,q$$

which is exactly the set of reduced equations specified in step 3.

The remaining steps need no formal proof. Thus this algorithm produces a reduced set of equations which have the same solution set as the original equations.

An example of this scheme is in order before its advantages and disadvantages are discussed. Consider the following equations

$$\begin{aligned}\bar{x}_1 + x_3 &= 1 \\ \bar{x}_1 + \bar{x}_2 &= 1 \\ \bar{x}_1 + \bar{x}_3 &= 1 \\ x_1 + x_2 &= 1 \\ \bar{x}_2 + x_3 &= 1\end{aligned}$$

First, eliminate x_1 . Note that the last equation does not involve x_1 and only one equation, the fourth, involves x_1 . Hence the reduced set is formed by forming the logical sum of the fourth equation with each of the first three to get the first reduced set:

$$\begin{aligned}\bar{x}_2 + x_3 &= 1 \\ x_2 + x_3 &= 1 \\ x_2 + \bar{x}_2 &= 1 \\ x_2 + \bar{x}_3 &= 1\end{aligned}$$

Note in this set that the third equation is trivial and can be deleted. Next eliminate the variable x_2 . All three equations involve either x_2 or its negation. Two involve the negation and one x_2 itself. Hence the second reduced set is the logical sum of the first and second and the first and fourth equations to yield:

$$\begin{aligned}x_3 + x_3 &= 1 \\ x_3 + \bar{x}_3 &= 1\end{aligned}$$

Of these two, the second is trivial. The first yields the value $x_3 = 1$ as the only solution. Substituting into the second reduced set yields $x_2 = 1$ as the only solution. Finally substituting into the first set gives $x_1 = 0$. Hence the unique solution is (0,1,1) which may be verified by exhaustion in this simple example.

Elimination of Redundant Equations

It was observed in the example above that the reduction process could produce redundant equations. This will not affect the solution but plays havoc with computer memory, in that extra equations can generate many redundant equations at each stage of reduction since

each equation is logically summed with many others. In practice this is intolerable, not only because of the extra computing time involved, but more importantly because of the extra computer main memory needed to store the equations. It is necessary to eliminate redundant equations at each stage. This may be done with the aid of the following.

Definition: Equation $a_1x + b_1\bar{x} = 1$ is said to imply the equation $a_2x + b_2\bar{x} = 1$ if each element of a_1 implies each element of a_2 and each element of b_1 implies each element of b_2 . This is equivalent to saying that if x_1 appears in the first equation it appears in the second equation and similarly for \bar{x}_1 . Terms may appear in the second equation which are not in the first equation however.

Theorem 2: Any equation which is implied by any other equation is redundant and may be deleted without affecting the solution to the set of simultaneous equations. This follows immediately by writing one equation as $A = 1$ and the equation which is implied by this one as $A+B = 1$. This is possible since every term in the first equation must be present in the second equation. Clearly if the first equation is satisfied, no constraint is added by the second for it is satisfied for all B . Hence it is redundant.

A more powerful reduction is provided by

Theorem 3: Consider two equations $x + A = 1$ and $x + B = 1$. If A implies B then these two equations are equivalent to

$$B = 1$$

$$x + A = 1$$

The proof follows from the consideration that $x + A = \bar{x} + B = 1$ has a solution only if $A+B = 1$. Since A implies B , it follows that $A+B = 1$ is equivalent to $B = 1$. This is evident from examination of the truth table for A and B adding the two conditions, A implies B and $A+B = 1$. Hence the second equation, $\bar{x} + B = 1$ reduces to $B = 1$ and the first equation is unchanged. This completes the proof.

Note that the effect of this theorem is to delete an equation involving \bar{x} . This means in the reduction, that equation passes immediately to the reduced set without being added logically to others and decreases significantly the number of equations produced.

The reduction procedure is now modified as follows:

1. Add to step 3: Check each new equation to determine if it contains both a variable and its negation. If so, delete it.
2. Add to step 1: Check each equation against the others using Theorems 2 and 3 to eliminate redundant equations.

With these changes, the number of equations generated is minimum and the solution easily generated. It should be noted that if the number of equations is large, it can be time consuming to compare each equations with all others to check for redundancy. However, the

computer implementation can be simplified greatly by defining the difference between two equations to be the number of variables which appear negated in the one equation but not the other. That is if x_1 appears in one equation and \bar{x}_1 in the other, this contributes to the value of the difference. If x_1 appears in both or only in one equation, it does not contribute to the difference. Then note that two equations can satisfy theorem 2 only if their difference is zero whereas they can satisfy theorem 3 only if their difference is 2. Adding a list structure to the implementation in memory permits the programs to keep track of which equations differ from others by 0 or 1, and only these need be checked to eliminate redundancy. This greatly decreases the computer time required for solution.

Relation of Boolean Functions to Walsh Functions

The Walsh Functions are related to Boolean expressions through the disjunctive normal form. The fundamental theorem of Boolean algebra states that any Boolean expression can be expressed in this form. Thus for any Boolean expression f , one can write

$$f = \sum_i B (m_i),$$

where \sum_B indicates the Boolean sum (inclusive disjunction) and m_i is an individual minterm. One can then form a truth table for the set of minterms. If one then Walsh transforms each minterm column and the column for the original function f , one finds that

$$W(f) = \sum_A [W(m_i)],$$

where $W(x)$ is the Walsh transform of x and \sum_A indicates the arithmetic sum.

For example, consider the Boolean expression

$$y = \bar{x}_1 x_2 + x_1 \bar{x}_2$$

The truth table representing this function would be

y	x_1	x_2	$\bar{x}_1 x_2$	$x_1 \bar{x}_2$
0	0	0	0	0
1	0	1	1	0
1	1	0	0	1
0	1	1	0	0

Taking the Walsh transform of the appropriate columns as indicated above yields

$$W(\bar{x}_1 x_2) = 1/4 (1 - 1, 1, -1)$$

$$W(x_1 \bar{x}_2) = 1/4 (1, 1, 1, -1)$$

$$W(y) = 1/4 (2, 0, 2, 0) = W(\bar{x}_1 x_2) + W(x_1 \bar{x}_2) \\ = \sum_A [W(m_i)]$$

Conversely, one can reduce a Boolean truth table to the disjunctive normal form in an equivalent way. This reduction is equivalent to solving the equation

$$f = \sum_B (c_i m_i)$$

for the values of the coefficients c_i , which will be 1 if m_i is an element of f and 0 is

it not.

For example, given the truth table

f	x_1	x_2
1	0	0
0	0	1
0	1	0
1	1	1

one can write the table with minterms as

f	$\bar{x}_1 \bar{x}_2$	$\bar{x}_1 x_2$	$x_1 \bar{x}_2$	$x_1 x_2$
1	1	0	0	0
0	0	1	0	0
0	0	0	1	0
1	0	0	0	1

The Walsh transforms of the columns are found using the Hadamard matrix

$$H_2 = \begin{bmatrix} + & + & + & + \\ + & - & + & - \\ + & + & - & - \\ + & - & - & + \end{bmatrix}$$

and are

$$W(\bar{x}_1 \bar{x}_2) = 1/4 (1, 1, 1, 1)$$

$$W(\bar{x}_1 x_2) = 1/4 (1, -1, 1, -1)$$

$$W(x_1 \bar{x}_2) = 1/4 (1, 1, -1, -1)$$

$$W(x_1 x_2) = 1/4 (1, -1, -1, 1)$$

and

$$W(f) = 1/4 (2, 0, 0, 2)$$

One can then write the linear equation

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 0 \\ 2 \end{bmatrix}$$

Since the matrix is orthogonal one can write immediately

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = 1/4 \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Thus c_1 and c_4 are = 1, and c_2 and c_3 are = 0, and

$$f = \bar{x}_1 \bar{x}_2 + x_1 x_2$$

This result was self-evident in the simple example but provides a simple basis for the computer implementation of expansion in minterms of large truth tables by solution of linear arithmetic algebraic equations.

The relation indicated above is entirely general. It may be stated as the theorem

$$(W(f) = \sum_A [c_i W(m_i)]) \Leftrightarrow (c_i = f_i)$$

The proof is straightforward

$W(f) = 1/2^n \sum H_n \{f_i\}$, where H_n is the Hadamard matrix.

The matrix of minterms M is merely the identity matrix

Therefore $HnM = Hn$

Accordingly there will be a vector \underline{c} such that

$$1/2^n H(\underline{c}) = 1/2^n H(f) \text{ iff } \underline{c} = \underline{f}$$

that is $c_i = f_i$.

We hoped initially that this relation would provide a means of rapid solution of the identification problem by permitting a transformation from simultaneous Boolean equations to simultaneous algebraic equations, and thus permit solution of n equations without the expansion to 2^n . However, this proved to be a false hope because the algebraic equations are in general not linear. Accordingly the Walsh transform per se does not appear particularly useful in the manipulation of Boolean models. However, in the course of this exploration, it was found that a related orthogonal transform is particularly useful. Its properties and some applications are described in the accompanying paper (6).

References

1. Chang, S. S. L. In Automatic and Remote Control Vol. 2 London, Butterworth, 1961, p. 712
2. Li, C.C. and J. Urquhart. Proc. 21st Ann Conf. Engr. Med. Biol. 10:29.5, 1968
3. Gann, D. S., L. E. Ostrander and J. D. Schoeffler In Systems Theory and Biology (Ed. M. D. Mesarovic) N. Y., Springer-Verlag 1968, p. 185.
4. Schoeffler, J. D., L. E. Ostrander and D. S. Gann Ibid, p. 201
5. Ostrander, L.E., J. D. Schoeffler and D. S. Gann in Data Acquisition and Processing in Biology and Medicine Vol. 5. Oxford, Pergamon 1968, p. 207.
6. Seif, F. J. and D. S. Gann. These Proceedings
7. Aveyard, R. A Theory for a Class of Discrete Event System. Ph.D Thesis. Case Western Reserve University, 1971.
8. Brand, J. E. Logical Modeling of Physiological Systems. Ph.D Thesis. Case Western Reserve University, 1971
9. Gann, D. S. Ann. N. Y. Acad. Sci. 156:740, 1969.
10. Gann, D. S. in Engineering Applications in Physiology, (Ed. J.H.U. Brown and D. S. Gann) N. Y. Academic Press, 1972.
11. Urquhart, J. and C. C. Li. Ann N. Y., Acad. Sci. 156:756, 1969.

MODELLING THE COMPOUND ACTION POTENTIAL OF THE NERVE

C. Bässwetter

Battelle-Institut e.V., Frankfurt/M., Germany

Abstract

In the paper the peripheral nerve is considered from a communications point of view as a special transmission system having dispersion-like properties. The mathematical model of the nerve trunk turns out to be a transversal filter with non-uniform distribution of the delay elements. It is shown that the input-output relationship of this special filter may be described in terms of discrete cross-correlation rather than of discrete convolution. The correlation matrix of the nerve is derived. The discrete impulse response of the system is shown to consist mainly of the fiber diameter histogram of the nerve. It follows that under the condition of non-overlapping unit discharge waveforms compound action potential and discrete impulse response become identical. Thus, at least in theory the fiber diameter histogram of a nerve usually determined by tedious electron-microscopic procedures may be measured electronically. The discrete model of the nerve may be used to compute the waveform of the compound action potential for arbitrary unit discharge waveforms, conduction distances and nerve types. The early graphical point-by-point procedure for reconstruction of the action potential is thereby replaced by a matrix algorithm well suited to be adapted to computer programming.

1. Introduction

Mammalian peripheral nerves consist of thousands of isolated individual fibers with different diameters and different conduction velocities. Thus, in general the actually recorded action potentials are compound, that means they contain the activities of a lot of individual neurons. Consequently it is difficult to separate the composite waveforms into a number of independent neuronal signal components. However, this has to be done before prostheses or other external devices controlled by neuronal information may be developed. In [5] an excellent review of all techniques currently being used is given. However, practically all methods are based on the severe restriction of non-overlapping unit discharge waveforms. As we shall see this condition is not met in most cases.

What we are looking for is the mathematical model of the nerve trunk as a transmission system based on the known physiological facts [4] thereby replacing the early graphical reconstruction technique [1][2][3] by analytical computation.

2. Statement of the Problem

Fig. 1 shows simplified a peripheral nerve containing many isolated individual fibers. At $t = 0$ a stimulation pulse is given. The amplitude is such that all fibers are innervated at once.

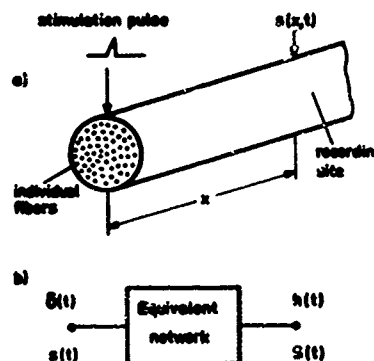


Fig. 1 a) Basic measurement scheme
b) Equivalent transmission system

Each fiber responds with the typical unit discharge waveform $s(t)$ traveling over the fiber with a velocity determined by the fiber diameter (and the nerve type). At x the compound action potential $S(x, t)$ is recorded. The unit discharge waveform of all fibers has the same shape, but not the same amplitude. It depends on the fiber diameter, too. The model of the nerve as transmission system is a linear system which may be described in the time domain by his impulse response $h(t)$. Evidently, the conduction distance will influence strongly the parameters of the system. These parameters and the input-output relationship of the system are to be determined.

3. Fundamentals of the Compound Potential Generation

All fibers of the gross nerve are grouped according to their diameters falling into the discrete classes having the mean diameter D_i . This

classification scheme is used by histologists to characterize a nerve by his fiber diameter histogram obtained by electron microscopic procedure. The histogram may be a function of the space variable x . However, little is known about this effect. Thus, at a first approximation it is neglected. Fig. 2 shows the typical shape of a fiber diameter histogram. N_i is the number of fibers falling into the i -th diameter class. The mean diameter of the i -th class is $D_i = i \cdot \Delta$ with $1 \leq i \leq n$ where Δ denotes the width of the classes.

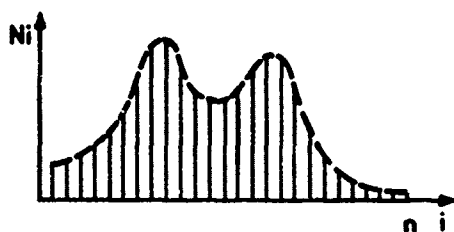


Fig. 2 Typical fiber diameter histogram

The unit discharge pulses of all fibers within a specific class can be represented by one spike whose amplitude is proportional to the diameter of the class. The delay of this spike is t_i . The conduction velocity of the N_i fibers of the i -th class is $v_i = k_0 D_i$. This linear relationship is true in a wide range of diameters. The delay t_i is

$$t_i = x/v_i = x/k_0 \cdot D_i = x/k_0 \cdot i \cdot \Delta = x/v_0 \cdot i = t_1/i \quad (1)$$

with the upper limit of the delay

$$t_1 = x/k_0 \cdot \Delta = x/v_0 \quad (2)$$

Thus, the influence of all fibers of the i -th class on the compound action potential is given by

$$s_i(t) = i \cdot N_i \cdot s(t - t_1/i) \quad (3)$$

where $s(t)$ denotes the actual unit discharge waveform of a fiber. The compound action potential is given by the superposition of all n signals representing the n diameter classes

$$S(t) = \sum_{i=1}^n i \cdot N_i \cdot s(t - t_1/i) \quad (4)$$

Considering this equation some general statements on the output signal $S(t)$ can be made (Fig. 3). If the fiber discharge signal has the finite length T the output waveform begins at $t = t_n$ and ends at $t = t_1 + T$. Thus the output signal of Equ. (4) is time limited, too, and has the duration

$$T_s = T + t_1(1 - 1/n) \quad (5)$$

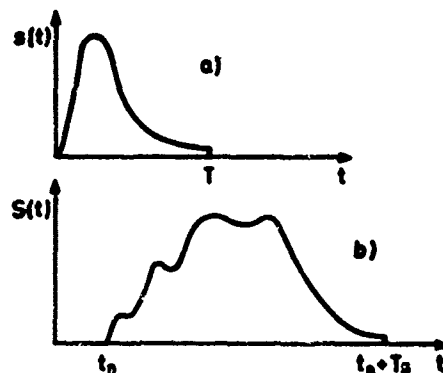


Fig. 3 a) Unit discharge waveform
b) Compound action potential waveform

The time delay is $t_n = t_1/n$. Full resolution occurs if the n -th and $(n-1)$ -th waveform don't overlap. Since the time difference is $t_{n-1} - t_n = t_1 \cdot n(n-1)$ we have the condition of non-overlapping $t_1 \geq T \cdot n(n-1)$. Using Equ. (1) this becomes

$$x \geq k_0 \cdot \Delta \cdot T \cdot n(n-1) \quad (6)$$

For conduction distances larger than given by Equ. (6) n individual neuronal spikes may be resolved. For real data usually being measured in physiology [3], saphenous nerve of the cat, $T = 1 \text{ ms}$, $v_0 = k_0 \cdot \Delta = 3.6 \text{ m/s}$, $\Delta = 0.5 \mu$ and $n = 16$ the minimum distance would be $x \geq 86 \text{ cm}$. This example shows that in general the compound action potential is caused by overlapping pulse trains since in many cases the minimum distance (Equ. (6)) is much larger than the overall size of the test animal.

4. The Peripheral Nerve as a Transversal Filter

Equ. (4) may be interpreted directly as a transversal filter (Fig. 4).

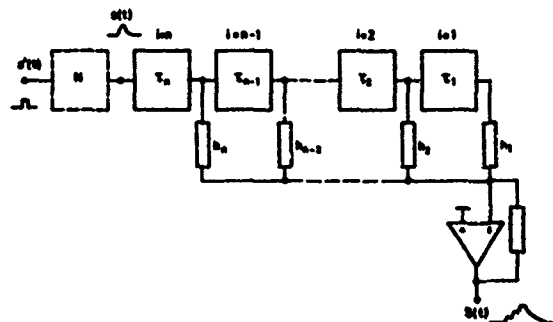


Fig. 4 Transmission model of the nerve

The network N at the front end of the filter transforms the quite arbitrary waveforms of the stimulation pulse into the actual wave shape of the fiber discharge spike $s(t)$. The weights of the filter are

$$h_i = i \cdot N_i \quad (7)$$

The time delays at the taps of the delay time are $t_i = t_1/i$. Thus the delay elements itself are $\tau_i = t_i - t_{i+1}$ or

$$\tau_i = t_1/i(i+1) \quad \text{with } 1 \leq i \leq (n-1) \quad (8)$$

Other than the ordinary transversal filter this specific type has a non-uniform time delay distribution. Independent of the length of the filter operator the first delay elements from the right are

$\tau_1 = t_1/2$ $\tau_2 = t_1/6$ $\tau_3 = t_1/12$ $\tau_4 = t_1/20 \dots$
The question is whether the input-output relationship of this filter is of the convolution type or the cross-correlation type. It has proved to be useful to analyze first the common discrete linear filter and the discrete cross-correlator. The differences to the discrete operation of the peripheral nerve may than easily be shown.

5. The Discrete Linear Filter

The input-output relationship of a linear system in the time domain is given by the convolution integral

$$S(t) = \int_{-\infty}^{\infty} s(\tau) \cdot h(t-\tau) d\tau = \int_{-\infty}^{\infty} s(t-\tau) \cdot h(\tau) d\tau \quad (9)$$

with the impulse response $h(t)$ of the time invariant system. For causality $h(t-\tau) = 0$ for $\tau > t$ holds. If $h(t)$ has the effective duration T the condition of causality becomes $h(t-\tau) = 0$ for $\tau > t - T$. Equ. (9) becomes

$$S(t) = \int_{t-T}^t s(\tau) \cdot h(t-\tau) d\tau = \int_0^T s(t-\tau) \cdot h(\tau) d\tau \quad (10)$$

Using the discrete variables $t = \gamma \cdot \Delta t$ and $\tau = \mu \cdot \Delta t$ with $\Delta t = \frac{T}{n}$ we have

$$S(\gamma \Delta t) = \sum_{\mu=1}^n s(\mu \Delta t) \cdot h[(\gamma - \mu) \Delta t] = \sum_{\mu=1}^n h(\mu \Delta t) \cdot s[(\gamma - \mu) \Delta t]$$

Writing the arguments as indices

$$S_{\gamma} = \sum_{\mu=1}^n s_{\mu} \cdot h_{\gamma-\mu} = \sum_{\mu=1}^n h_{\mu} \cdot s_{\gamma-\mu} \quad (11)$$

The ordinary convolution is a commutative operation. Input signal and

impulse response are interchangeable. Equ. (11) denotes also a matrix operation. Using the causality $h_{\gamma-\mu} = 0$ for $\mu > \gamma$ the convolution matrix \underline{H} becomes for the first expression in (11)

$$\begin{pmatrix} h_1 & 0 & 0 & 0 & \dots & 0 \\ h_2 & h_1 & 0 & 0 & \dots & 0 \\ h_3 & h_2 & h_1 & 0 & \dots & 0 \\ h_4 & h_3 & h_2 & h_1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ h_n & h_{n-1} & h_{n-2} & h_{n-3} & \dots & h_1 \end{pmatrix} = \underline{H}$$

The discrete convolution is expressed as

$$\underline{S} = \underline{H} \cdot \underline{s} \quad (12)$$

where \underline{S} denotes the output signal vector and \underline{s} the input signal vector. The n -th row of the convolution matrix is the discrete filter operator $\{h_n\}$ characterized by equally spaced samples of the impulse response.

Considering the right expression in Equ. (11) a different matrix notation of the discrete convolution results

$$\underline{S} = \underline{s} \cdot \underline{h} \quad (13)$$

The transformation matrix is now

$$\begin{pmatrix} s_1 & 0 & 0 & 0 & \dots & 0 \\ s_2 & s_1 & 0 & 0 & \dots & 0 \\ s_3 & s_2 & s_1 & 0 & \dots & 0 \\ s_4 & s_3 & s_2 & s_1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ s_n & s_{n-1} & s_{n-2} & s_{n-3} & \dots & s_1 \end{pmatrix} = \underline{s}$$

The discrete convolution according to (12) or (13) is performed by the transversal filter with n sections (Fig. 5).

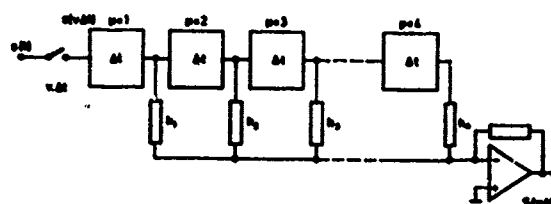


Fig. 5 Discrete model of convolution

The discrete impulse response of the filter is the operator $h = (h_1, h_2, h_3, \dots, h_n)$. Considering a time-limited input signal $s(\gamma \Delta t)$ and $n=4$ the complete set of equations determining the output signal $S(\gamma \Delta t)$ becomes

$$\begin{aligned}
S_1 &= s_1 h_1 \\
S_2 &= s_1 h_2 + s_2 h_1 \\
S_3 &= s_1 h_3 + s_2 h_2 + s_3 h_1 \\
S_4 &= s_1 h_4 + s_2 h_3 + s_3 h_2 + s_4 h_1 \\
S_5 &= s_2 h_4 + s_3 h_3 + s_4 h_2 \\
S_6 &= s_3 h_4 + s_4 h_3 \\
S_7 &= s_4 h_4
\end{aligned}$$

With Equ. (12) and (13) in mind this set of equations may be interpreted as two different matrix operations. The time delay Δt of the output signal is not included.

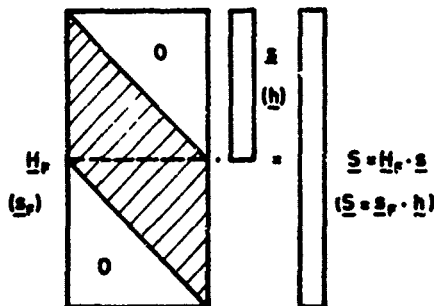
The first notation

$$\begin{pmatrix} h_1 & 0 & 0 & 0 \\ h_2 & h_1 & 0 & 0 \\ h_3 & h_2 & h_1 & 0 \\ h_4 & h_3 & h_2 & h_1 \\ 0 & h_4 & h_3 & h_2 \\ 0 & 0 & h_4 & h_3 \\ 0 & 0 & 0 & h_4 \end{pmatrix} \cdot \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{pmatrix} = \begin{pmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \end{pmatrix} \quad (14)$$

and the second

$$\begin{pmatrix} s_1 & 0 & 0 & 0 \\ s_2 & s_1 & 0 & 0 \\ s_3 & s_2 & s_1 & 0 \\ s_4 & s_3 & s_2 & s_1 \\ 0 & s_4 & s_3 & s_2 \\ 0 & 0 & s_4 & s_3 \\ 0 & 0 & 0 & s_4 \end{pmatrix} \cdot \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{pmatrix} = \begin{pmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \end{pmatrix} \quad (15)$$

or graphically for both



Equ. (14) and (15) follow from the fact that also the discrete convolution forms a commutative operation, that is $H_F \cdot s = s^T \cdot h$. It can be seen that the transformation matrices H_F or s^T of the discrete filter realizable as transversal filter are rectangular ones. Thus, only in the special case of square matrices the corresponding inverse discrete filter is realizable.

6. The Discrete Cross-Correlator

For continuous signals $s(t)$ convolution and cross-correlation are linked by

$$CCF(\theta) = s(t) * h(-t) \Big|_{t=\theta}$$

Cross-correlation is identical to convolution by the time-reversed filter operator. Since this is true also for discrete signals and operators the physical realization of the discrete cross-correlation is again the transversal filter. However, the elements of the operator $h_k = (h_n, h_{n-1}, h_{n-2}, \dots, h_2, h_1)$ are located in reversed sequence. Corresponding to chapter 5 the set of equations determining the elements of the output vector \underline{S} may be interpreted as two different matrix multiplications. For the previous example ($n=4$) the first operation becomes $H_F^T \cdot \underline{S} = \underline{S}$

$$\begin{pmatrix} h_4 & 0 & 0 & 0 \\ h_3 & h_4 & 0 & 0 \\ h_2 & h_3 & h_4 & 0 \\ h_1 & h_2 & h_3 & h_4 \\ 0 & h_1 & h_2 & h_3 \\ 0 & 0 & h_1 & h_2 \\ 0 & 0 & 0 & h_1 \end{pmatrix} \cdot \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{pmatrix} = \begin{pmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \end{pmatrix} \quad (16)$$

and the second

$$\begin{pmatrix} 0 & 0 & 0 & s_1 \\ 0 & 0 & s_1 & s_2 \\ 0 & s_1 & s_2 & s_3 \\ s_1 & s_2 & s_3 & s_4 \\ s_2 & s_3 & s_4 & 0 \\ s_3 & s_4 & 0 & 0 \\ s_4 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{pmatrix} = \begin{pmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \end{pmatrix} \quad (17)$$

The discrete cross-correlation is a commutative operation also, that is $H_F^T \cdot s = s^T \cdot h$. It should be noted here that the commutative property of convolution and cross-correlation is not self-explanatory but needs the condition of operators be represented by equally spaced samples. As chapter 4 shows is not the case for the peripheral nerve.

7. The Peripheral Nerve As a Specific Cross-Correlator

The physical model of the transmission properties of the peripheral nerve is the transversal filter having non-uniform time delay distribution (Fig. 4). The system responds to a unit impulse at $t=0$ after the time delay $t_n = t_1/n$ with $h = (h_n, h_{n-1}, h_{n-2}, \dots, h_2, h_1)$ where the time interval between two samples h_i and h_{i-1} is $t_i = t_1/(i-1)$ with $2 \leq i \leq n$. The duration of the impulse response $T_n = t_1(1-1/n)$ approaches t_1 for $n \rightarrow \infty$ (Fig. 6). Evidently the time-reversed fiber histogram (Equ. (7) and Fig. 2)

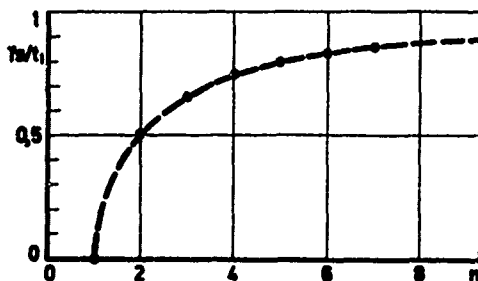


Fig. 6 Duration of the discrete impulse response

multiplied with the weights i plays the role of the discrete impulse response. For the simple case of uniform distribution $N_i = \text{const.}$ the weights of the filter h_i have a triangular shape. Since the relationship between delay and index i is non-linear the resultant impulse response is represented by non-equally spaced samples (Fig. 7). The maximum delay t_1 is a function of the conduction distance x (Equ. 2). Thus there is a linear change in scale and the shape of the impulse response is not affected.

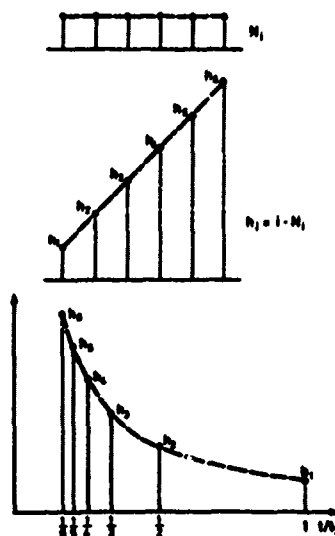


Fig. 7 Discrete impulse response of the nerve model ($n=6$)

It would follow from this that it must be possible to measure the fiber diameter histogram of a nerve bundle electronically by stimulating the nerve with Dirac impulses and recording the impulse response thereby eliminating the time consuming electron-microscopic techniques. However, all fibers of a nerve bundle are able only to propagate the typical unit discharge waveforms $s(t)$, usually called "spike". This is equivalent to the fact that the input of the real transversal filter (Fig. 4)

is inaccessible. The only input experimental accessible is the input of the pulse forming network N . Nevertheless it may be possible to determine the fiber histogram of a nerve from the compound action potential $S(y\Delta t)$, that is the output of the model for unit discharge input $s(t)$, if the condition (6) for non-overlapping spikes is met. Beyond this limit compound action potential and discrete impulse response become identical. For the example of Fig. 7 the limit of resolution is given by $x = 3,6 \frac{m}{\mu s} \cdot \text{ms} \cdot 30 \approx 10 \text{ cm}$. With this minimum conduction distance $n=6$ separated spike waveforms may be observed. Dividing the discrete amplitudes by $i=1,2 \dots 6$ yields the fiber histogram. This new technique just briefly mentioned here has still to be verified experimentally.

Evidently, the transmission properties of the nerve may be described as a specific cross-correlator rather than as operation of the convolution type since the impulse response of the system is the time-reversed filter operator. Additionally, it is easy to see that the transmission behavior of the nerve cannot be formulated as a matrix operation like Equ. (16). That is why there exists no shifting increment for operators defined by unequally spaced samples such that a notation like Equ. (16) would be possible. Therefore, the notation of Equ. (17) must be used. In the following the cross-correlation matrix s_k in general shall be derived.

8. The Cross-Correlation Matrix of the Nerve Trunk

With the maximum time delay component $t_1 = x/v_0 = m \cdot \Delta t$ and eliminating the minimum delay component $t_n = (m/n) \cdot \Delta t$ we obtain from Equ. (4) with $t = y \cdot \Delta t$

$$S[(y-m/n)\Delta t] = \sum_{i=1}^n i \cdot N_i \cdot s[(y(\frac{n-i}{n})\Delta t)]$$

Using the abbreviation for the shifting increment $r_i = \frac{(n-i)m}{n \cdot i}$ (18)

it follows

$$S_{y-m} = \sum_{i=1}^n h_i \cdot s_{y-r_i} \quad (19)$$

In general the shifting parameter r_i is non integer. The elements s_{y-r_i} are the samples of the unit discharge waveform taken at the times $t = (y-r_i)\Delta t$. If the unit spike $s(t)$ of the fibers has the time duration $T_p \cdot \Delta t$ the time length of the compound potential $S(y\Delta t)$ becomes (Equ. (5))

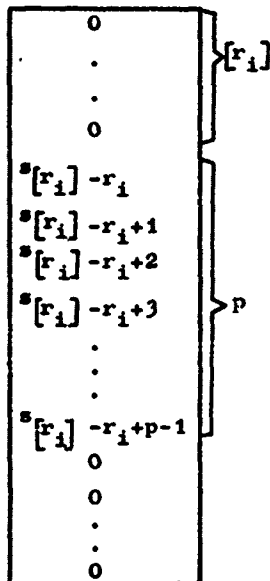
$$T_s = [p + m(1 - \frac{1}{n})] \Delta t = x \cdot \Delta t \quad (20)$$

The number of rows of the correlation matrix \underline{s}_k will therefore be $z+1$ or $[z]$ depending on whether z takes integer values or not. $[z]$ denotes in the latter case the next integer number following z . It is useful to consider the correlation matrix column by column. We obtain the y -th element of the i -th column vector $s(y-r_i)\Delta t =$

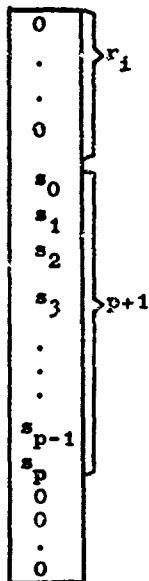
$$= \begin{cases} s([r_i] - r_i + j)\Delta t & \text{for } [r_i] - p \leq [r_i] + p - 1 \\ s(j\Delta t) & \text{for } r_i - p \leq r_i + p \\ 0 & \text{for } 0 \leq [r_i] - 1 \\ 0 & \text{for } 0 \leq r_i - 1 \end{cases} \quad (21)$$

with $j=0,1,2 \dots p$. The upper row is valid for non-integer values of r_i , the lower for integer values. Thus the i -th column vector of the matrix \underline{s}_k has the following structure:

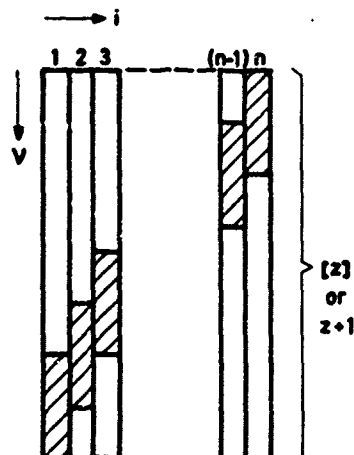
r_i non integer



r_i integer



The complete cross-correlation matrix has then the form:



The construction procedure of the correlation matrix shall be explained by a simple example.

Example: $p=3, n=4, m=6$

from Equ. (20) $z=7.5$ thus $[z]=8$

from Equ. (18) we obtain the shifting parameters

$$\begin{aligned} r_1 &= 4.5 \rightarrow [r_1] = 5 \\ r_2 &= 1.5 \rightarrow [r_2] = 2 \\ r_3 &= 0.5 \rightarrow [r_3] = 1 \\ r_4 &= 0 \end{aligned}$$

Regarding Equ. (21) the complete matrix may be constructed

$$\begin{matrix} & \rightarrow i \\ \downarrow y & \begin{bmatrix} 0 & 0 & 0 & s_0 \\ 0 & 0 & s_{1/2} & s_1 \\ 0 & s_{1/2} & s_{3/2} & s_2 \\ 0 & s_{3/2} & s_{5/2} & s_3 \\ 0 & s_{5/2} & 0 & 0 \\ s_{1/2} & 0 & 0 & 0 \\ s_{3/2} & 0 & 0 & 0 \\ s_{5/2} & 0 & 0 & 0 \end{bmatrix} \end{matrix} = \underline{s}_k$$

The simple example shows the unequally spaced samples of $s(t)$ forming the rows of the matrix. The vector of the compound action potential represented by equally spaced samples $S(y\Delta t)$ is then obtained by the matrix operation $\underline{S} = \underline{s}_k \cdot \underline{h}$ where \underline{h} denotes as earlier the filter operator representing the fiber diameter histogram multiplied by the weights i . It may easily be shown that compound action potential \underline{S} and impulse response $\underline{h}_k = (h_4, h_3, h_2, h_1)$ become identical if the conduction distance is beyond the limit given by Equ. (6). If x increases than the number of the rows $[z]$ or $z+1$ increases, too. For the example the limit for non-overlapping spikes would be $m \geq pn(n-1) = 12$ for $p=1$ and $z=10$. Choosing $m=15$ yields $z=12.25 \rightarrow [z]=13$. The shifting increments are in this case:

$$\begin{aligned} r_1 &= 11.25 \rightarrow [r_1] = 12 \\ r_2 &= 3.75 \rightarrow [r_2] = 4 \\ r_3 &= 1.25 \rightarrow [r_3] = 2 \\ r_4 &= 0 \end{aligned}$$

The correlation matrix \underline{s}_k thus becomes

$$\begin{bmatrix} 0 & 0 & 0 & s_0 \\ 0 & 0 & 0 & s_0 \\ 0 & 0 & s_{3/4} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & s_{1/4} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ s_{3/4} & 0 & 0 & 0 \end{bmatrix} = \underline{s}_k$$

Multiplying the matrix by the filter operator \underline{h} yields the discrete impulse response \underline{h}_k being represented by the four unequally spaced samples h_4, h_3, h_2 , and h_1 .

Acknowledgements

The work covered by this paper was performed under a Battelle-funded project. Thanks are due to Miss U. Spychala and Miss H. Staeger for typing the manuscript and for help in correcting the grammar, and Dr. A. Holz for helpful discussions.

References

1. Erlanger, J., Gasser, H.S., Bishop, G.H.: The compound nature of the action current of nerve as disclosed by the cathode ray oscillograph, Amer. J. Physiology, Vol. 70 (1924) No. 3, pp. 624-666
2. Gasser, H.S., Erlanger, J.: The role played by the sizes of the constituent fibers of a nerve trunk in determining the form of its action potential wave, Amer. J. Physiology, Vol. 80 (1927) No. 3, pp. 522-547
3. Gasser, H.S., Grundfest, H.: Axon diameters in relation to the spike dimensions and the conduction velocity in mammalian A fibers, Amer. J. Physiology, Vol. 127 (1939) pp. 393-414
4. Dug, F.T.: The length and diameter of the node of Ranvier, IEEE Trans. on Bio-Medical Engineering, Vol. BME-17 (1970), No. 1, pp. 21-24
5. Glaser, E.M.: Separation of neuronal activity by waveform analysis, In: Advances in Biomedical Engineering, Vol. 1, Academic Press, 1971
6. Schmidt, E.M.: Unit activity from peripheral nerve bundles utilizing correlation techniques, Med. & Biol. Engineering, Vol. 9 (1971), pp. 665-674
7. Olson, W.H., Bement, S.L.: Compound action potentials: Computer analysis of parameters that determine the waveform, Proc. 23rd Annual Conference of Engineering in Medicine and Biology, 1970, Washington, D.C., p. 46

AN APPLICATION OF WALSH FUNCTIONS TO THE MONITORING OF ELECTROCARDIOGRAPH SIGNALS

P. J. Milne, Department of Electrical Engineering
and the Institute for Environmental Research
Kansas State University, Manhattan, Kansas 66502

N. Ahmed, Depts. of Electrical Engineering and Computer Science
Kansas State University, Manhattan, Kansas 66502

R. R. Gallagher, Dept. of Electrical Engineering
Kansas State University, Manhattan, Kansas 66502

S. G. Harris, Dept. of Surgery and Medicine
Kansas State University, Manhattan, Kansas 66502

Abstract

A plausible method for classification of electrocardiograph data as coming from a normal or an abnormal subject using the Walsh-Hadamard Transform is demonstrated. Several types of ECG signals were obtained from research canines. These signals were declared normal or abnormal by the veterinary cardiologist at Kansas State University's Dykstra Veterinary Hospital. The Walsh-Hadamard power spectrum of these signals was then obtained and four of these spectral points were used to train a specific pattern classifier. The results of using the classifier show perfect classification of the normal and abnormal signals from a given subject. In the case of signals from a mixed population, a population consisting of samples from all subjects, ten spectrum points were used to train the classifier which proved to be eighty-nine per cent correct.

The results suggest that the Walsh-Hadamard power spectrum could prove useful in characterizing ECG's for the purpose of automatic classification. Thus, recommendations for future work along these lines are included. Several applications of these results are suggested.

Introduction

Many applications of the principles of engineering to medicine have been made in recent years. One such application is the use of the digital computer to process the vast amounts of medical data available [1, 2]. The use of the computer to aid in diagnosis comes as a result of its ability to handle this data.

One of the signals available for processing is the electrocardiograph signal (ECG). This signal is generated by the heart muscle during the cardiac cycle. The signal is

measured using a lead system, which is a particular placement of measurement electrodes on the surface of the body. The shape and frequency of this signal provides the cardiologist with information pertaining to the physical well-being of the heart. The computer can then be used to distinguish between normal and abnormal ECG signals. It is used most commonly for mass screening and for patient monitoring. In mass screening the goal is to automatically detect abnormalities in large populations. In individual monitoring a single patient is monitored and any future changes in his ECG can then be detected.

Past work has emphasized the time domain approach [1, 2]. With this approach the parameters a cardiologist looks for in the time domain are measured and compared to a range of known characteristics for normal. This method has probably been the most popular since the parameters it uses are those the cardiologist ordinarily examines.

Frequency analysis, however, can be an important tool when used with the computer. Past work has shown that in many cases abnormalities are accompanied by an increase in the high-frequency content of the ECG. One such frequency analysis is the method using the Walsh-Hadamard Transform (WHT) [3].

Fundamentals of Electrocardiography

The electrocardiogram is a measure of the electrical activity of the heart. This electrical signal is generated during the cardiac cycle by the depolarization and repolarization of the heart muscle cells during their process of contraction and relaxation. The actual potential measured across the cell membrane is due to an ionic gradient. The changes in this ionic gradient occur with the muscle action of the heart and these changes in the ionic potential are recorded as the ECG. (See Fig. 1).

The heart is a cyclic pump. The cardiac cycle includes pumping blood from the ventricles to the body and to the lungs as well as the return of blood from the body and the lungs to the heart. The right and left ventricles are the pumping chambers of the heart with the right sending oxygen deficient blood to the lungs and the left sending oxygen-laden blood to the body.

The atria are the receiving chambers of the heart with the left receiving oxygen-laden blood from the lungs and the right receiving oxygen deficient blood from the body.

The ECG under most circumstances may be assumed to be periodic since it is generated during the cardiac cycle. The sino-atrial node initiates the stimulus for the contraction of the heart muscle. The stimulus travels across the atria causing them to contract and then after a short delay in passing through the atrio-ventricular node it passes down the septum and on through the ventricles which then contract. The depolarizations of the atria and of the ventricles are evidenced by the P wave and the QRS complex respectively as shown in Figure 1. After these muscle cells contract they return to their initial state through the process of repolarization. The repolarization of the atria is masked by the QRS complex while the repolarization of the ventricles produces the T wave of the ECG.

The ECG's waveform is dependent upon the recording electrodes' placement on the body. This arrangement is called a lead system. The lead system used in this study was the McFee orthogonal lead system. It consisted of 10 leads and a ground. This lead system theoretically measures the ECG along 3 mutually orthogonal axes as illustrated in Figure 2.

The ECG is an important adjunct used by the cardiologist in diagnosing heart ailments. It gives information used in the diagnosing of such conditions as myocardial infarction, and many systemic diseases affecting the heart. Any aid to the interpretation of the ECG is thus an aid to the prevention of death from heart disease.

As previously suggested, most past efforts in automating the interpretation of the ECG have focused on the time domain approach. This is perhaps due to the fact that cardiologists are familiar with and can attach significance to the time domain signal. The frequency characteristics of the ECG have been used mainly for specifying the recording and monitoring equipment necessary for time domain analysis. In this study the frequency approach appeared to be valuable also in distinguishing between normal and abnormal ECG's.

The Walsh-Hadamard Transform

The Walsh-Hadamard Transform (WHT) or Bifore (Binary Fourier Representation) is an orthogonal transformation in which square waves form the basis set. These square waves are analogous to the sine waves of the Fourier Transform. The one dimensional Walsh-Hadamard Transform is defined as

$$\{B_x(n)\} = \frac{1}{N} [H(n)] \{X(n)\} \quad (1)$$

where $n = \log_2 N$

$\{B_x(n)\}$ is an $(NX1)$ vector whose components $B_x(k)$, $k = 0, 1, \dots, (N-1)$ are the transform coefficients,

$[H(n)]$ is an $N \times N$ Hadamard matrix,

$\{X(n)\}$ represents the sampled values of an ECG in the form of an $(NX1)$ vector.

The power spectrum corresponding to the above components is defined as [3]

$$P_0 = B_x^2(0) \quad (2)$$

$$P_s = \sum_{k=2^{s-1}}^{2^s-1} B_x^2(k)$$

$$s = 1, 2, \dots, n$$

$$n = \log_2 N.$$

This power spectrum possesses two useful properties. (1). The spectral points P_i , $i=0, 1, \dots, \log_2 N$ are invariant to shifts of a sampled ECG signal $X(m)$; and (2). They also represent the distribution of power in the ECG.

The WHT used for the case of two channels of ECG data as shown in Figure 3 is defined as follows:

$$[B(n_1, n_2)] = \frac{1}{N_1 N_2} [H(N_1)] [X(n_1, n_2)] [H(N_2)] \quad (3)$$

$$n_1 = \log_2 N_1$$

$$n_2 = \log_2 N_2$$

$[B(n_1, n_2)]$ is an $(N_1 \times N_2)$ transform matrix

$[X(n_1, n_2)]$ is an $(N_1 \times N_2)$ data matrix

and $[H(k)]$ is a $(2k \times 2k)$ Hadamard matrix.

In the application of interest (see Fig. 3), $N_1 = 2$ and $N_2 = 32$. Thus the two-dimensional WHT in (3) can be expressed in terms of the one-dimensional WHT of the "sum" and "difference" channels defined by $X(k) + Y(k)$, $k = 0, 1, \dots, 31$ and $X(k) - Y(k)$, $k = 0, 1, \dots, 31$ respectively.

The WHT power spectrum is related to the discrete Fourier power spectrum in the following manner.

$$\begin{aligned}
 p_0 &= C_x^2(0) \\
 p_1 &= C_x^2\left(\frac{N}{2}\right) \\
 p_s &= 2 \sum_{k=0}^{s-2-1} \left| C_x[2^{n-s}(2k+1)] \right|^2 \quad (4)
 \end{aligned}$$

$$s = 2, 3, \dots, n$$

$$n = \log_2 N$$

and $C_x(i)$, $i = 0, 1, \dots, N$ are the discrete Fourier Transform power spectrum points.

ECG Data Acquisition

The intent of this study was to demonstrate the feasibility of automatically classifying ECG's using the WHT. The experimental data was obtained from a group of laboratory canines. The canine was chosen because of the similarity of its heart and ECG to that of the human. This similarity would then allow the extension of the techniques developed to the human.

The signals were recorded at Kansas State University's Dykstra Veterinary Hospital with the system shown in Figure 4. The procedure used in recording the data included anesthetizing the canine, recording its normal ECG and then inducing various cardiac abnormalities while continuing to record its ECG. The resulting signals were then interpreted by a veterinary cardiologist and classed as normal, abnormal or questionable.

One segment of the ECG signal was chosen upon which to distinguish the classes of ECG's. This segment was the QRS segment which, as stated earlier, corresponds to the depolarization and contraction of the ventricles. Figure 3 shows the two channels of the sampled QRS segment used. Not all abnormalities can be detected in this segment but it was felt that enough of them could be to warrant using it as a first step in this study.

The sampling of the ECG began with the start of the QRS complex and ended 80 milliseconds later after having recorded 32 samples. Consequently, the corresponding frequency analysis applies to the frequency interval of $0 < F \leq 200 \text{ Hz}$. The X and Y leads of the McFee orthogonal lead system were sampled simultaneously to obtain a (2×32) matrix of data points which was then used to compute the two-dimensional WHT power spectrum. The resulting spectrum consisted of 12 WHT points as follows:

$$\begin{array}{llll}
 P(0,0) & P(0,1) & \dots & P(0,5) \\
 P(1,0) & P(1,1) & \dots & P(1,5)
 \end{array}$$

Classification Considerations

A training algorithm which uses a least-squares mapping technique was used [5, 6]. The basic idea used was to map (in the least squares sense) the training samples of a class k , $k = 1, 2, \dots, K$ into a unit vector V_k in a K -dimensional decision space. All the components of V_k are zero except for the k^{th} one which is unity. The corresponding mapping matrix is obtained during the training process. Then, the trained classifier assigns an incoming pattern to class i_0 if the pattern is mapped closest to the unit vector V_{i_0} in the decision space.

Discussion of Results

Figure 5 shows the results obtained using different numbers of components of the power spectrum. These results were obtained using three classes for training; normal, abnormal and questionable. The abnormal and questionable classes were then combined into one class and the efficiency was calculated using these two classes. For this case the signals from the canines studied were mixed. As can be seen, the efficiency of classification increases to 89% when 10 components are used. Fig. 6 shows the groupings used from 4-10 WHT power spectrum points.

A related measure of the success of classification involves the measurement of the number of abnormal signals classed as normal. When this type of error occurs the individual in question might not receive the medical care he should have. Figure 7 shows the frequency of this type of error versus the number of components used to classify. As can be seen the best results are obtained for the case of 10 components. The other type of error, that of a normal classed as abnormal, is not as serious, since the individual would ordinarily seek further medical care and other tests would show him to be normal.

The classification process was then applied to detect variations from normal within a given canine. The power spectrum points of the canine's normal were compared with that of its induced abnormalities using a two class version of the classifier mentioned above. With four components used the classifier was 100% efficient.

Concluding Remarks

The results of this study suggest three plausible uses for the techniques developed. First, in patient monitoring a normal or acceptable signal would be obtained from the individual being monitored. Then variations from this normal could be detected and could

alarm the appropriate medical personnel to the change in the patient's condition. This application is strongly supported by the success in separating normal from abnormal within a given canine.

A second, related, application would be for use in serial electrocardiography. In serial electrocardiography an individual's normal ECG is recorded for comparison with one recorded at a later time. Any changes detected would again be called to the attention of appropriate medical personnel.

Mass screening is the third application. In this case large numbers of normal and abnormal ECG's would be collected for the training of the classifier. The resulting classifier would then be used to detect abnormal ECG's in large populations. The success of the classifier in separating normal as shown in figure 5 and abnormal in the mixed population supports this application.

This study has shown the feasibility of using the Walsh-Hadamard Transform approach to detect cardiac disease.

REFERENCES

- [1] J. Wartak: Computers in Electrocardiography. Springfield, Illinois. Charles C. Thomas Publication, 1970.
- [2] C. A. Caceres and L. S. Dreifus: Clinical Electrocardiography and Computers. New York, New York. Academic Press Inc., 1970.
- [3] N. Ahmed, et. al.: Bifore or Hadamard Transform. IEEE Transactions on Audio and Electroacoustics. AU 19: 225-234, 1971.
- [4] N. Ahmed and R. M. Bates: A Power Spectrum and Related Physical Interpretation for the Multi-Dimensional Bifore Transform. Applications of Walsh Functions Proceedings: 45-50, 1971.
- [5] N. Ahmed: Personal Communication. Kansas State University, Manhattan, Kansas.
- [6] W. G. Chaplin and V. S. Levadi: A Generalization of the Linear Threshold Decision Algorithm to N-Dimensions. The Second Symposium on Computers and Information Sciences, Columbus, Ohio, August, 1968.

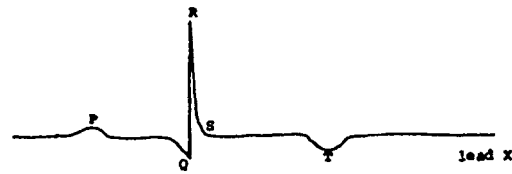


Fig. 1 The Electrocardiograph Signal as Recorded on Lead X.

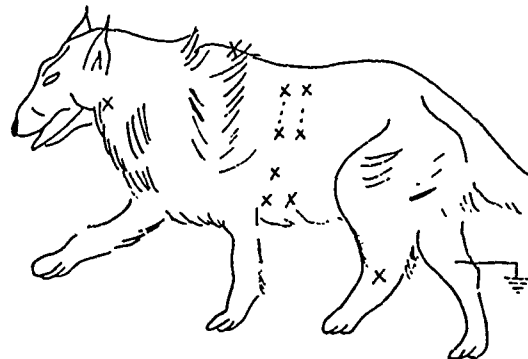


Fig. 2 The Placement of Electrodes on the Canine Using the McFee Lead System.

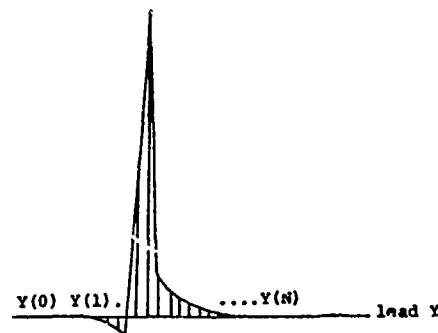
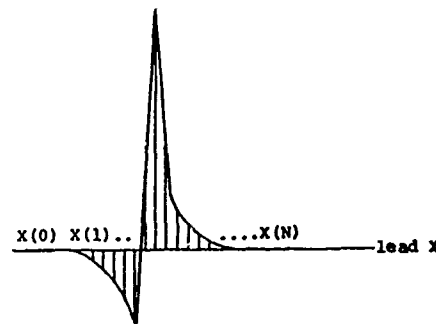


Fig. 3 The sampled QRS Interval as Recorded With the McFee Lead System on Leads X and Y. $N = 31$.

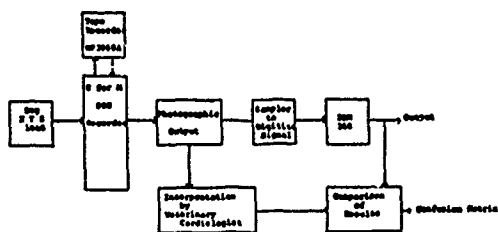


Fig. 4 The Data Acquisition System Used for Electrocardiograph Classification.

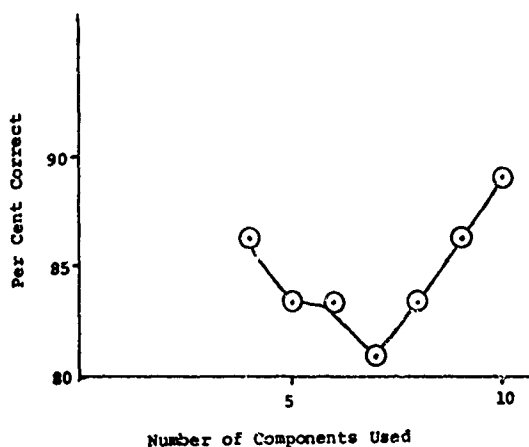


Fig. 5 The Success of the Classifier Shown as a Function of the Number of WHT Power Spectrum Points Used.

4 Components: P(0,4), P(0,5)
P(1,4), P(1,5)

5 Components: P(0,3) P(0,4) P(0,5)
P(1,4) P(1,5)

6 Components: P(0,3) P(0,4) P(0,5)
P(1,3) P(1,4) P(1,5)

7 Components: P(0,2) P(0,3) P(0,4) P(0,5)
P(1,3) P(1,4) P(1,5)

8 Components: P(0,2) P(0,3) P(0,4) P(0,5)
P(1,2) P(1,3) P(1,4) P(1,5)

9 Components: P(0,1) P(0,2) P(0,3) P(0,4)
P(0,5) P(1,2) P(1,3) P(1,4)
P(1,5)

10 Components: P(0,1) P(0,2) P(0,3) P(0,4)
P(0,5) P(1,1) P(1,2) P(1,3)
P(1,4) P(1,5)

Fig. 6 The Grouping of Components Used to Classify ECG Signals.

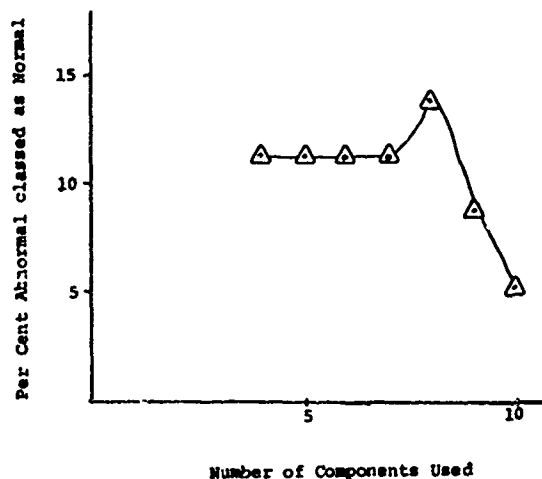


Fig. 7 The Per Cent Abnormal Classified as Normal as a Function of the Number of WHT Power Spectrum Points Used. This is the More Serious Type of Error.

HEART RATE REPRESENTATION USING WALSH FUNCTIONS

C. W. Thomas & A. J. Welch

The University of Texas at Austin

Introduction

The common interpretation of the term "heart rate" is the number of heart beats per minute measured by feeling the pulse within an artery, listening to heart sounds or looking at an electrocardiogram. In each case, the heartbeats are counted for a minute and that number is called the heart rate or pulse rate. Although heart rate is considered a relatively stable measurement, there is a large variation in the times between heart beats. The interval between beats may be determined from the electrocardiogram as shown in Figure 1(a). The reciprocal of this interval is designated as instantaneous heart rate. In this paper, we view heart rate as a piecewise, continuous function whose value at any time is the instantaneous heart rate (See Figure 1(b)). The average value of the continuous heart rate function over one minute is the classical heart rate. This general interpretation of heart rate is necessary if we are interested in the dynamic effects of body temperature, respiration, emotional state, etc., upon the beating of the heart.

Variations of the heart intervals are more medically and physiologically significant than the mean heart rate or even the wave shape of the electrocardiogram. This is obviously true in abnormal heart conditions in which extra beats or missing beats are common symptoms. In the normal human, a large variation of heart intervals with respiration is usually indicative of good health. Therefore, a more general representation of heart rate than the one minute averages may be clinically useful. In many clinical situations, especially intensive care or coronary care units, most medical and physiological research, and even astronaut monitoring, the electrocardiogram is routinely monitored and the times of occurrence of heart beats are available for analysis.

Representation of Heart Rate

Womack (1) has attempted to extract respiration information from instantaneous heart rate by frequency analysis procedures. In order to work with a bandlimited system, the discrete level heart rate function of Figure 1(b) was represented as a continuous function as shown in Figure 1(c). This is a convenient representation even though it is not directly measurable.

A representation we wish to consider in detail is a sequence of impulses. The time of occurrence of each impulse corresponds to the time of the heartbeat (R wave peak of the electrocardiogram). That is, the occurrence of heartbeats may be represented as:

$$I = \sum_i \delta(t - T_i) \quad (1)$$

where T_i is the time of occurrence of the heart beat.

Definition of Heart Rate Using Walsh Transform

In the above representation of heart rate, the time of occurrence of each heartbeat is marked by an impulse. If I is the input to a flip-flop, the output of the device would be a two-level signal which may be mathematically represented by a sequence of unit step functions as:

$$H = -1 + 2 \sum_i [\mu(t - T_{2i-1}) - \mu(t - T_{2i})] \quad (2)$$

The instantaneous heart rate is determined from the time between zero crossings. The rate of heart beats is represented by the rate of zero crossings of the heart rataplan function defined by equation 2.

If the heart rate is constant, the generated time function will be a square wave. If the heart rate varies, the zero crossings of the square wave will vary and the instantaneous heart rate is the instantaneous rate of zero crossings of the heart rataplan function.

To represent the rate of zero crossings, we periodically sample the heart rataplan function (or the state of the flip-flop) for the desired time period and take the Walsh Transform of the sampled function. The sequence coefficients from the Walsh Transform represent the heart rate function.

There are several advantages to this representation of heart rate. First, the time function, i.e. the heart rataplan function, is simple to generate with a computer or with a flip-flop. Second, it can be sampled uniformly allowing flexibility in the choice of time period and the number of samples in that period. Third it allows straightforward computation of the Walsh Transform coefficients which represent heart rate.

Experimental Data

In the normal human and many other animals, the most prominent heart rate variation is the sinus arrhythmia, i.e. the variation in heart rate with respiration as described by Richardson et al (2). In this paper we use four sets of data in which the sinus arrhythmia is clearly present. The first two sets are simulations of square wave and sine wave variations in heart intervals. The third set is data taken from a subject who was breathing sinusoidally as described in Womack (1). In the fourth set of data, the respiration is unknown, but the stage of sleep is known from other considerations as described by Welch (3).

In the two sets of simulated data a constant rate of 72 beats per minute (intervals of 5/6 second) is modulated according to

$$T_i = T_{i-1} + 5/6 + (M/72) \text{ sign } Q \quad (3)$$

and

$$T_i = T_{i-1} + 5/6 + \frac{M}{72} Q \quad (4)$$

where $Q = \sin\left(\frac{2\pi R}{60} T_{i-1}\right)$

M = amplitude of the interval modulation

and R = respiration rate in breaths per minute

In equation 3 the sinus arrhythmia is simulated as a square wave modulation of the heart intervals, while equation 4 simulates a sinusoidal sinus arrhythmia. In both cases, the amplitude of the sinus arrhythmia is represented by M . When $M = 0$, the heart rate is constant, i.e., our heart rataplan function is a square wave. For non-zero values of M , the heart interval in equation 4 vary sinusoidally between $5/6 - M/72$ and $5/6 + M/72$ or the heart rate varies between

$$\frac{72}{1 + M/60} \quad \text{and} \quad \frac{72}{1 - M/60}$$

Notice that the variation in heart rate is not exactly sinusoidal since the mapping from heart intervals to heart rate is nonlinear. We have assumed that sinusoidal sinus arrhythmia means sinusoidal variation of heart intervals not sinusoidal variation of heart rate.

In equation 3, non-zero values of M yield a square wave modulation of heart intervals. While this type of sinus arrhythmia is not experimentally possible, it does furnish a comparison with the sinusoidal case which can be closely approximated in the laboratory.

Simulated data was obtained using equations 3 and 4 with $M = 4$ and 8 and $R = 4, 6, 8, 10, 12$, and 14 breaths per minute. The mean in-

terval was 5/6 seconds which corresponds to a rate of 72 beats per minute. One minute segments were used to construct the heart rataplan function which was sampled at 1024 per minute.

The Walsh Transform of the sampled function from the square wave variation was calculated for each one minute segment and the Walsh Transform coefficients plotted in Figure 2. As in frequency modulation of sinusoids, increased modulation amplitude M , resulted in an increase in number of major coefficients around the fundamental.

The effect of increasing the frequency of the modulation, i.e. increasing the simulated respiration rate, is not so easily described and appears to be frequency dependent. The transform has fewer major coefficients for rates of 8 and 12 than for the other rates. However the coefficients obviously vary with the modulation frequency.

The Walsh Transform of the simulated sinusoidally varying heart intervals are shown in Figure 3. Again the number of significant coefficients varies with both amplitude and frequency of the sinusoidal modulation.

The data from the human subject breathing sinusoidally was used to construct one minute segments of heart rataplan functions whose Walsh Transforms are shown in Figure 4. The frequency of the sinusoids are the same as in the simulated data in Figure 3. However, the simulation was sinusoidally varying heart intervals while the data from the human subject was sinusoidal respiration.

The Walsh Transform of the human data in Figure 4 is similar to both sets of simulated data in Figure 2 and 3. The human data is better correlated with the simulated data with the larger modulation amplitude. The plots in Womack (1) show a heart rate variation of about 15 beats per minute. The simulated data with $M = 8$ has about the same heart rate variation. Therefore, the correlation with the human data should be higher than for $M = 4$.

The fourth set of data was obtained during a normal night of sleep. Respiration in the sleeping human is not sinusoidal and its amplitude and rate vary with sleep stage.

The heart rataplan function was constructed, sampled, and Walsh Transformed for the sleep data as in the previous sets of data. The heart rate functions plotted in Figure 5 obviously change with sleep stage, but the respiration rates cannot be determined by visual comparison with Figures 2 and 3.

This is not surprising since the variation in heart intervals is certainly not sinusoidal or square wave, and as shown in Figures 2 and 3, the number of significant coefficients varies with modulating frequency. The non-sinusoidal non-symmetric modulation of the heart intervals during sleep seems to increase the complexity of the heart rate function.

To compare the Walsh Transform of the heart rate function with the Fourier Transform of the same function, both transforms of the sleep data are shown in Figure 6. First the coefficients from both transforms are plotted, then the square root of the sum of the squares of the two coefficients at each frequency and frequency are plotted.

The latter plots represent power at bands of frequency and frequency. In other words, they are the amplitude spectra as defined by Harmuth (4).

Conclusions

The representation of heart rate using the Walsh Transform has been presented. The data shows that variations in heart rate are represented by several coefficients in the transform. The number of significant coefficients depends on the amplitude, period, and complexity of the modulating function.

We have not solved any problem, but we have demonstrated a tool which may be useful in problems involving the use of heart rate and especially variance of heart rate. We have used only one time period and one sampling rate, but these two parameters may be chosen to fit a particular problem. Such flexibility and the computational speed make this technique potentially useful.

References

1. Womack, B. F. Analysis of Respiratory Sinus Arrhythmia Using Spectral Analysis and Digital Filtering. *IEEE Trans. BME*, Vol. 18, No. 6, Nov. 1971.
2. Richardson, P. C., Welch, A. J., and Groshner, R. G. Heart Rate Response to Square Wave Breathing. *Aerospace Medicine*, Vol. 42, No. 9, Sept. 1971.
3. Welch, A. J. Period Analysis of Space Flight EEG. *Aerospace Medicine*, Vol. 42, No. 6, June 1971.
4. Harmuth, H. F. A Generalized Concept of Frequency and Some Applications. *IEEE Trans. Info. Th.*, Vol. 14, No. 3, May 1968.

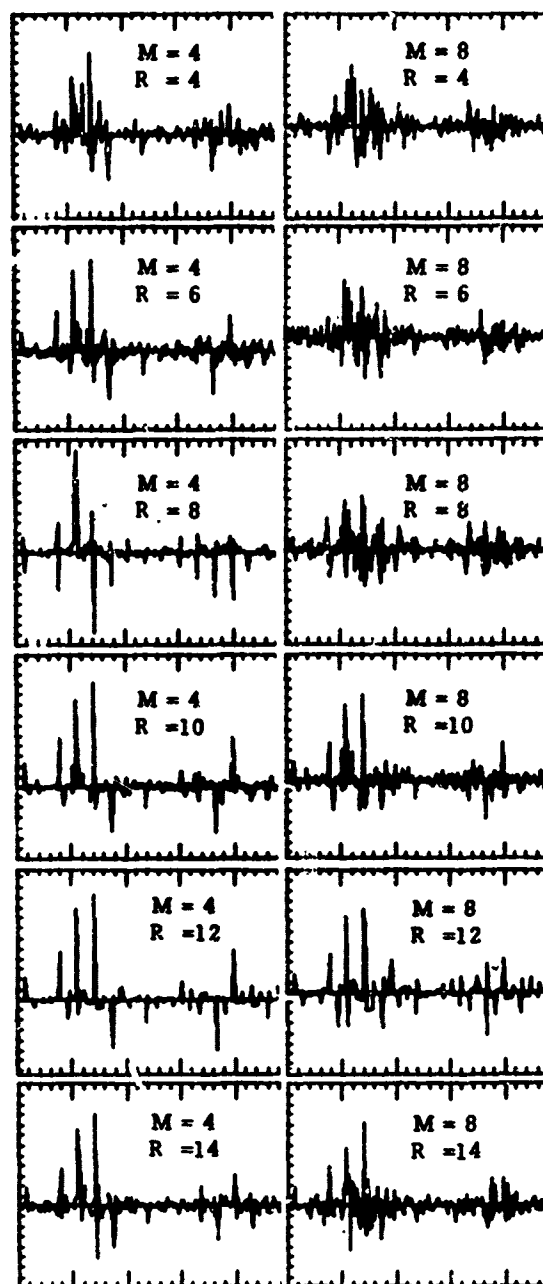


Figure 2. Heart rate functions from the simulated square wave variation of heart intervals. The frequency scale is one to 120 zpm and the vertical scales are the same for all plots.

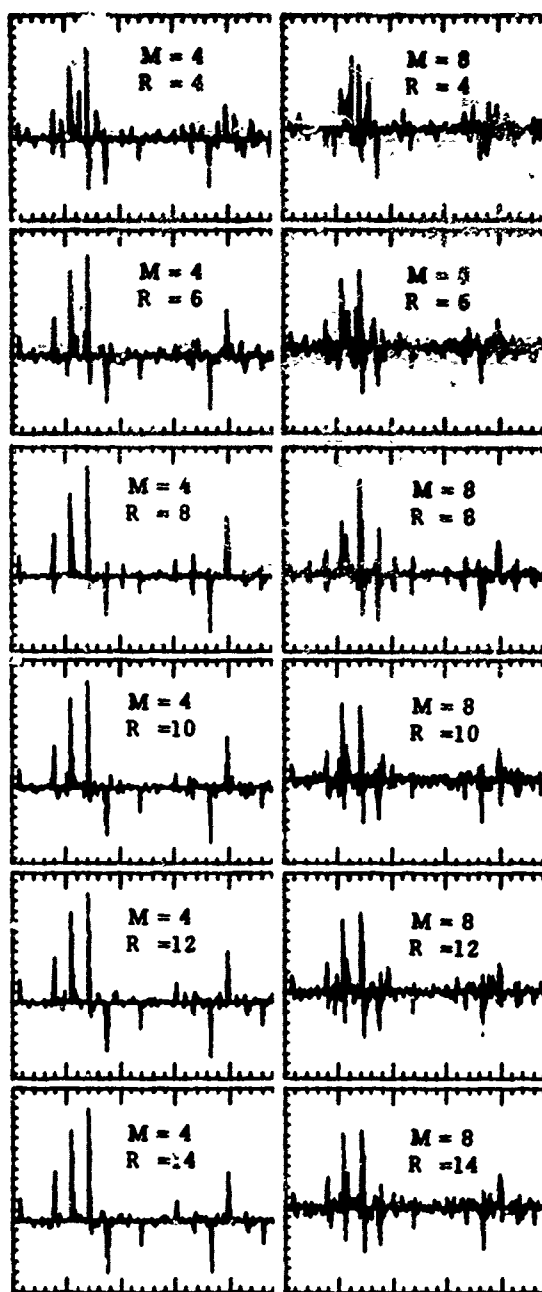


Figure 3. Heart rate functions from the simulated sine wave variation of heart intervals. The sequence and vertical scales are the same as in Figure 2.

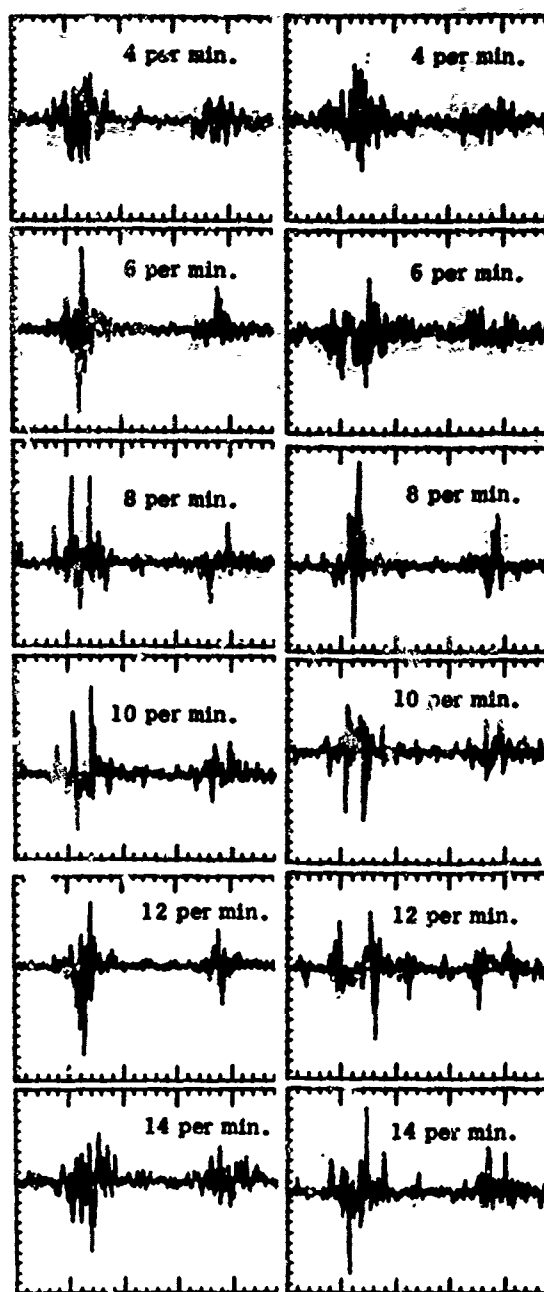


Figure 4. Heart rate functions for the two human subjects breathing sinusoidally. The breathing rates and the plot scales are the same as in Figures 2 and 3.

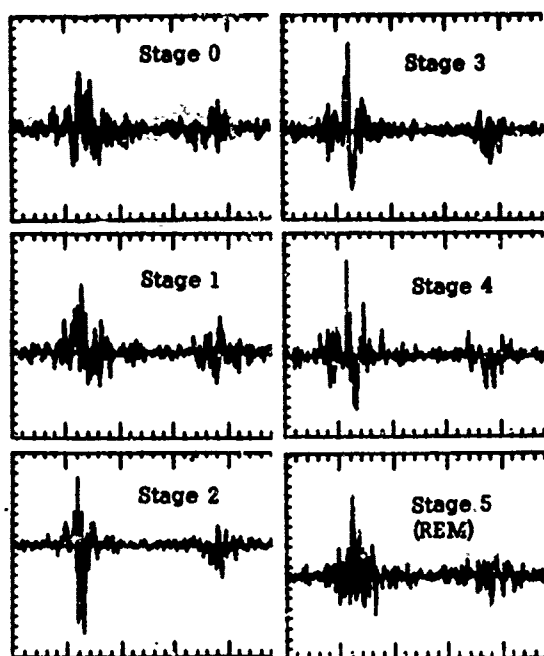


Figure 5. Heart rate functions for the sleeping human in different stages of sleep. The plot scales are the same as in the previous three figures. Stage 0 is the awake but drowsy state, stages 1, 2, 3, and 4 are progressively deeper sleep, and stage 5 is at the level of stage 1 but with REM (rapid eye movement)

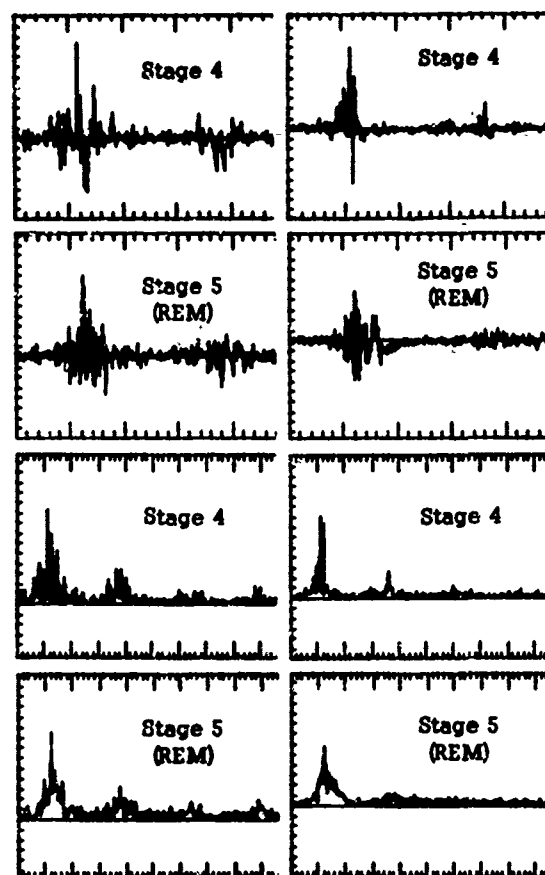


Figure 6. Comparison of the Walsh Transform (left) and the Fourier Transform (right) of the heart rate function for the sleeping human in stages 4 and 5. The upper plots are coefficients from the transform, while the lower plots are the amplitude spectra. The scales for the upper plots are the same as in the four previous figures. In the amplitude spectra plots, the frequency scale is 1 to 240 zpm.

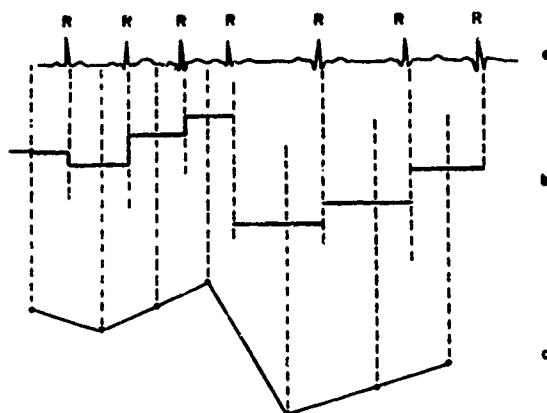


Figure 1. Definition of heart rate from the electrocardiogram.

WALSH POWER SPECTRA OF HUMAN ELECTROENCEPHALOGRAMS

W. C. Yeo and J. R. Smith
Department of Electrical Engineering
University of Florida
Gainesville, Florida

Introduction

Brain wave (EEG) analysis is a very important tool in many research and clinical applications. The clinical diagnosis of epilepsy and sleep research are two areas which owe their development almost entirely to the discovery of the EEG. These studies always entail the processing of large amounts of data, and have prompted many investigators to try automated techniques to assist in the data analysis and in attempts to discover new knowledge about the EEG.

Fourier spectral analysis is the technique most widely applied. Applications include the analysis of EEG data collected from astronauts [1] and the discrimination among states of consciousness (sleep and wakefulness) [2]. The utilization of the EEG to classify sleep into several stages [3] has led to the development of automated systems for the computer classification of sleep stages [2, 4, 5].

This study was carried out to determine if the Walsh Transform could be utilized in discriminating between sleep stages and to compare Walsh and frequency EEG spectra.

The human sleep EEG, as shown in Figure 1, consists of six sleep stages (stage W, 1, 2, 3, 4, and REM). The sleep stage W (wakefulness)

is characterized by α -wave activity (8 - 12 Hz nearly sinusoidal activity), and stage 1 by a relatively low voltage, mixed frequency (2 - 7 Hz) EEG. Stage 2 is defined by presence of phasic activities of approximately 1 second duration (sleep spindles and/or K-complexes) superimposed on a background of relatively low voltage, mixed frequency EEG activity [3]. Stage 3 and 4 contain moderate and large amounts (in time) of high amplitude, .5-2.5 Hz activity respectively. REM sleep - thought to coincide with periods of dreaming - has an EEG similar to sleep stage one, but is distinguished by rapid eye movements and or the attenuation of the electromyogram.

Methodology

Three human sleep EEG's were recorded on magnetic tape and then processed off-line. All the data was obtained from the frontal position (F1-F7) of the skull, except for the awake data (stage W) which was obtained from the occipital region (O3-OZPZ). The data presented here were obtained from a single night's reading of a 22 year old male with a relatively low amplitude EEG.

The EEG data was filtered by a low pass filter with an upper cut-off frequency of 25 Hz

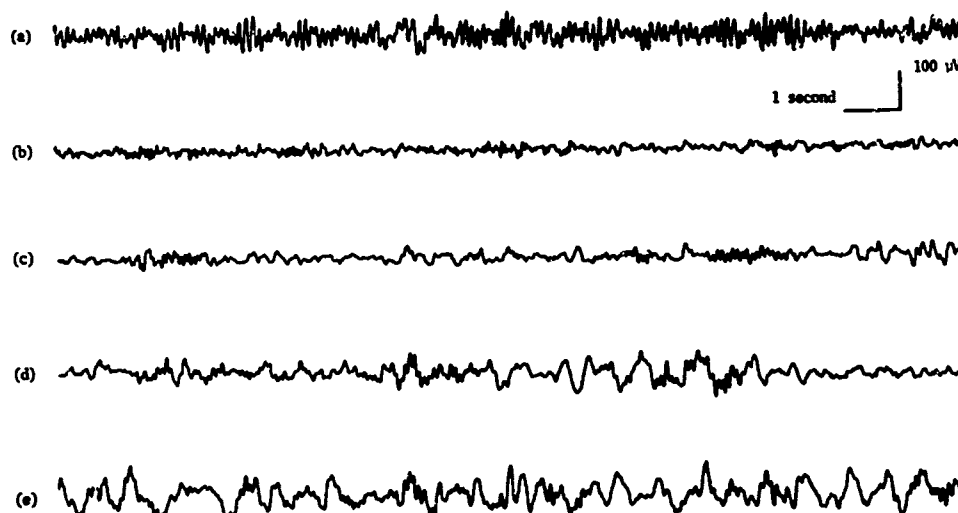


Figure 1. Human sleep EEG
(a) stage W (b) stage 1 (c) stage 2
(d) stage 3 (e) stage 4.

and then converted to digital data and processed with a PDP-8I computer. The sampling period was 32 seconds and the sampling rate was 32 samples/sec., giving a maximum frequency of 16. This frequency limit is close to the upper frequency limit of interest in sleep EEG data.

The input data was transformed into a series of Walsh functions [6] using the Fast Bifore Transform to solve the equation

$$X = H x$$

where x and X are 1024 element column vectors representing the input and transformed data respectively, and H a matrix generated by a periodic sampling of Walsh functions [7].

Figure 2(a) shows the computation algorithm employed to obtain the Walsh Transform. The data processing differs from Pratt et al's method [8] in that the number of blocks, each indicated by a rectangle, and the number of data points in each block on a level are first calculated to obtain terminal conditions for the loops used in computation of intermediate data on next level. All the intermediate data on the next level are then processed through a nested loop and the results stored in separate memory locations. The loop for computation of intermediate data in odd blocks has, as a unit operation, only the addition of two data while the loop for even blocks consists of two operations as shown in Figure 2(b). For intermediate data storage an additional 1024 memory locations are allocated. This computation algorithm has the advantage that the Walsh coefficients are obtained in frequency order with

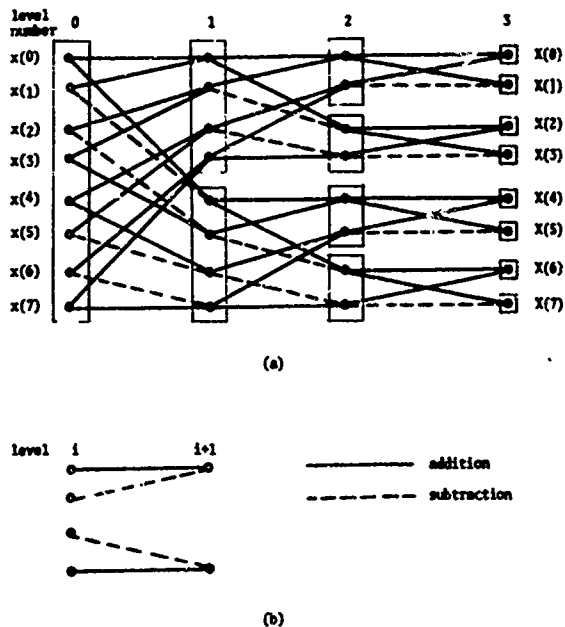


Figure 2. Data Processing Algorithm.

fewer computing operations. The instructions needed for the transformation occupy only about 100 memory locations and can be applied to any number of input data points which are of two's power. The transformed Walsh coefficients were squared and summed pairwise for each frequency. The spectra were converted to a logarithmic scale and displayed on a storage oscilloscope, from which the Walsh Spectra plots have been obtained.

The same epochs were analyzed with a General Radio 1921 Real Time Analyzer to obtain Fourier power spectra estimates. The analyzer consists of 45 filters, spaced 1/3 octave apart, covering the frequency spectrum 3.5 Hz to 80k Hz. The Analyzer samples the outputs from the filters to estimate the Fourier power spectra. The EEG was reproduced at 32 times real speed, given an equivalent filter bandwidth of 3.5/32 Hz to 2.5k Hz. An integration time of 1 second was employed - corresponding to 32 seconds real time. 19 spectral estimates in the frequency range (.3-19.7 Hz) have been plotted.

Results and Discussion

Figure 3 shows typical Fourier spectra and Walsh spectra of each sleep stage. It is observed, in general, that the Walsh power spectra is more diffuse throughout the frequency range than the corresponding Fourier power spectra. The better waveform discrimination of Fourier power spectra for EEG activity of nearly sinusoidal form is illustrated in Figures 3(A0) and 3(B0). The approximately 10 Hz alpha activity is observed in the Walsh spectra 3(A0), but subharmonics are more prominent than in the corresponding Fourier spectra. Aliasing is not believed to contribute to the observed frequency diffusion as the Fourier power spectra shows relatively low EEG activity above 16 Hz. Also, ten seconds epochs (approximately 100 samples/sec.) showed the same frequency diffusion. A comparison of the plots for the other sleep stages show that the activities present in the frequency plots are also evident in the frequency plots. However the peaks are more pronounced in the frequency plots, undoubtedly due to the sinusoidal like characteristics of all sleep stages except 1 and 1-REM.

Lubin et al [2] have described the inability frequency spectra to discriminate between sleep stages using stepwise multiple regression. The same problem exists if Walsh spectra are used. Figure 4 illustrates the problem by showing the similarity in the Walsh spectra of a sleep stage 2 and a sleep stage 3. Larsen and Walter [4] reported that a multiple discriminant function analysis of frequency spectra could discriminate between sleep stages if an adequate training sample was used. Such techniques would probably be as effective using Walsh spectra.

Similar characteristics were observed in the comparison of the spectra of the other two subjects.

Conclusions

The discrimination by Walsh power spectra is not as good as the Fourier spectra for sinusoidal like EEG data, but activity present in the frequency spectra was evident in the Walsh spectra plot. The Walsh power spectral analysis has very significant advantages over Fourier spectral analysis in that the former requires much less computing time and computer memory, advantages which are quite significant in a laboratory possessing a mini-computer. For investigators desiring to make a spectral decomposition of EEG data, the poorer discrimination of nearly sinusoidal waveforms may be more than compensated by the ease with which the Walsh spectra can be computed.

References

1. Adey, W. R., Kado, R. T., and Walter, D. O., Computer Analysis of EEG Data from Gemini Flight GF-7, *Aerospace Medicine*, 1967, 38: 345-359.
2. Lubin, A., Johnson, L. C., and Austin, M. T. Discrimination Among States of Consciousness using EEG Spectra, *Psychophysiology*, 1969, 6: 122-131.
3. Rechtschaffen, A. and Kales, A. (eds.): *A Manual of Standardized Terminology, Techniques and Scoring Systems for Sleep Stages of Human Subjects*, Public Health Service, U. S. Government Printing Office, Washington, D. C., 1968.
4. Larsen, L. E. and Walter, D. O., An Automatic Method of Sleep Staging by EEG Spectra, *Electroenceph. Clin. Neurophysiol.*, 1970, 28: 459-467.
5. Smith, J. R. and Karacan, I., EEG Sleep Staging by an Automatic Hybrid System, *Electroenceph. Clin. Neurophysiol.*, 1971, 31: 231-237.
6. Walsh, J. L., A Closed Set of Normal Orthogonal Functions, *Am. J. Math.*, 1923, 55: 5-24.
7. Ahmed, N., Rao, K. R., Addussettar, A. L., BIFORE or Hadamard Transform, *IEEE Transactions on Audio and Electroacoustics*, 1971, 19: 225-234.
8. Pratt, W. K., Kane, J., Andrews, H. C., Hadamard Transform Image Coding, *Proceedings of the IEEE*, 1969, 57: 58-68.

Acknowledgement: This research was supported by NSF grant GK 15373 and NIMH grant 16960.

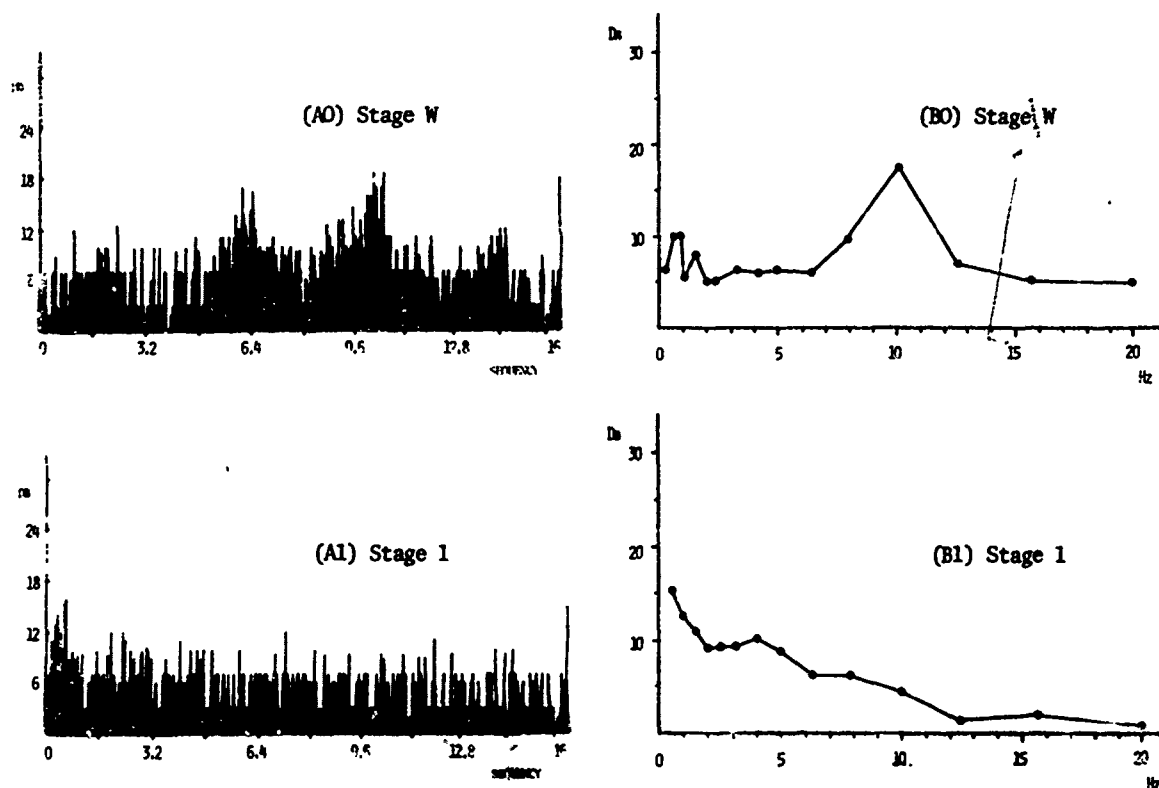


Figure 3. Walsh and Fourier power spectra of Human Sleep EEG (A: Walsh, B: Fourier)

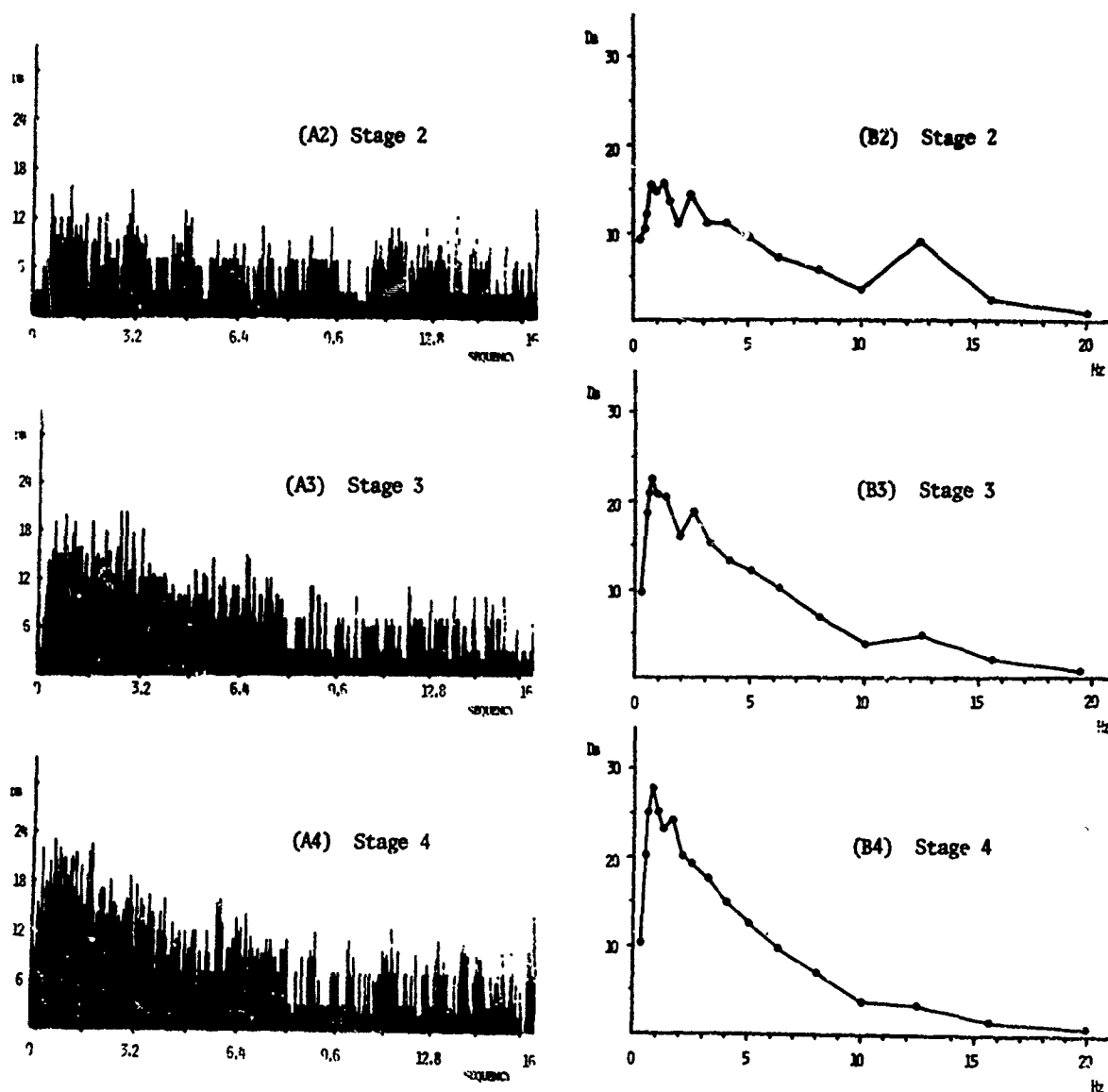


Figure 3 (Continued)

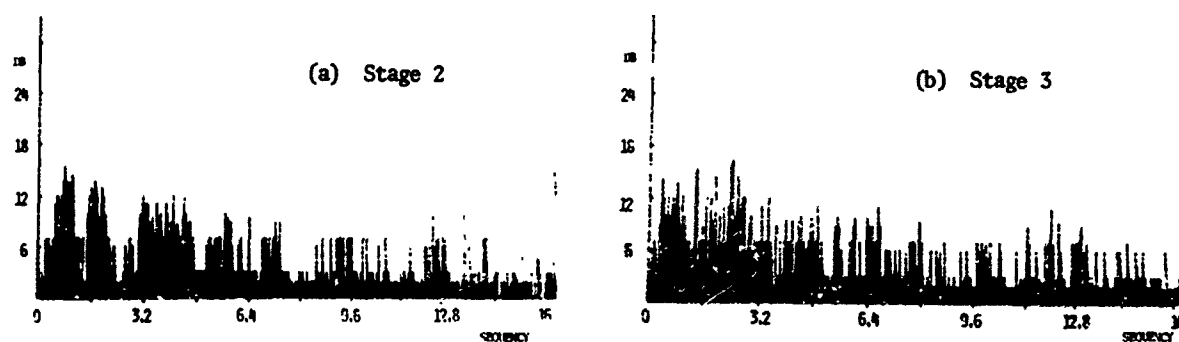


Figure 4. Similar Walsh Spectra from Different Sleep Stages.

SPEECH PROCESSING WITH WALSH FUNCTIONS

by

H. Gethöffer

Technische Hochschule Darmstadt

Fachbereich Nachrichtentechnik

Darmstadt, West-Germany

Introduction

A main topic in data transmission especially for speech exists in pointing out optimal source and channel coding [1,2]. The basic block diagram is shown in figure 1.

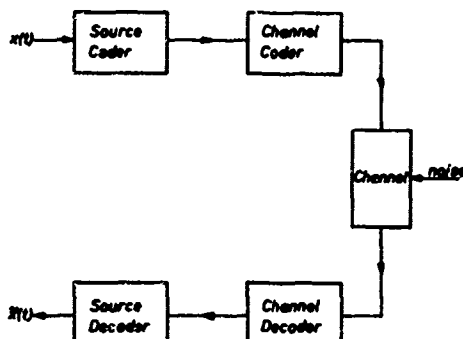


Figure 1. Basic communication system

The redundancy of the source signal is suppressed in the source coder and the reduced binary data is protected against noise and disturbance in the channel coder. The channel coding and decoding may be separately solved and we now consider a subsystem as shown in figure 2.

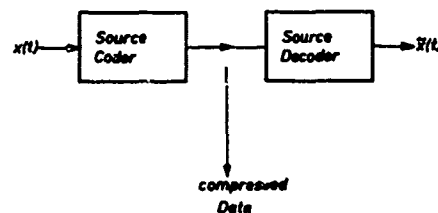


Figure 2. Source coding and decoding

Because speech signals exist one part of nearly periodic waveforms, the voiced vowels, and on the other hand of the more unregular unvoiced parts optimal coding must refer to these characteristics in speech. This segmentation will cause difficulties-especially the pitch detection problem-and first we make no difference between voiced and unvoiced signals.

Linear orthogonal transformation offers a method for data compression by means of coding the spectral coefficients. The spectral distribution is exploited by different quantization schemes. These speech processing systems are also known as orthogonal transform vocoders [3].

Linear Transformation

A time-continuous frequency limited signal $x(t)$ can be represented with respect to the sampling theorem as a discrete series of samples $x(k\Delta t)$. The discrete signal samples are written in finite intervals as column vectors

$$\begin{aligned}\underline{x}(0) &= (x_0, x_1, \dots, x_k, \dots, x_{N-1})' \\ \underline{x}(1) &= (x_N, x_{N+1}, \dots, x_{N+k}, \dots, x_{2N-1})' \\ &\vdots \\ \underline{x}(\nu) &= (x_{\nu N}, x_{\nu N+1}, \dots, x_{(\nu+1)N-1})'\end{aligned}$$

and transformed with any regular matrix \underline{G} with the elements $g_{j,k}$ on a generalized spectral domain

$$\hat{\underline{x}}(\nu) = \underline{G} \underline{x}(\nu)$$

$\hat{\underline{x}}(\nu)$ is the spectral vector of the ν -th interval and has the elements

$$\hat{\underline{x}}(\nu) = (\hat{x}_0^\nu, \hat{x}_1^\nu, \dots, \hat{x}_j^\nu, \dots, \hat{x}_{N-1}^\nu)'$$

The time dependent coefficients may be regarded as samples of convolution functions between signal and the rows of the transform matrix [4]. The inverse transform onto the original domain is given by

$$\underline{x}(\nu) = \underline{G}^{-1} \hat{\underline{x}}(\nu)$$

There is a one-to-one correspondance between signal and transform domain and no errors are made by linear transformation [5].

Most signals are correlated because they are output-functions of convolution type systems. Therefore the optimum linear transformation is given by the Loève-Karhunen expansion because the correlation between the signal samples is eliminated in the transform domain.

The statistics of the signal are given in terms of correlation coefficients ρ_i defined as

$$\rho_i = \frac{r_{xx}(i)}{r_{xx}(0)}$$

and r_{xx} denotes the autocorrelation of the stationary process. The covariance matrix of the process is of Toeplitz form and given by

$$\text{Cov}(\underline{x}) = \sigma^2 \begin{bmatrix} 1 & \rho_1 & \rho_2 & \dots & \rho_{N-1} \\ \rho_1 & 1 & \rho_1 & \dots & \rho_{N-2} \\ \rho_2 & \rho_1 & 1 & \dots & \rho_{N-3} \\ \vdots & & & \ddots & \vdots \\ \rho_{N-1} & \dots & \dots & \dots & 1 \end{bmatrix}$$

$\sigma^2 = r_{xx}(0)$ is the variance of the process. The Loève-Karhunen transform diagonalizes the covariance matrix [6]

$$\underline{K} \text{Cov}(\underline{x}) \underline{K}^{-1} = \text{Diag}(\lambda_i)$$

and eliminates the correlation of the process in the transform domain. For wide-sense stationary processes and $N \rightarrow \infty$ the covariance-matrix is diagonalized by the discrete Fourier matrix. But for finite N and non-stationary signals as short speech sample blocks the optimal finite Loève-Karhunen transforms are given by

$$\underline{K}(\nu) \text{Cov}(\underline{x}(\nu)) \underline{K}(\nu)^{-1} = \text{Diag}(\lambda_i(\nu))$$

and will be different in each interval. For each interval the Eigenvalues and Eigenvectors must be determined which requires a large computation time and it is nearly impossible to perform real-time processing of the signal. There are other methods [7,8] based on a mean Karhunen-Loève expansion where the signal is treated as a stationary process.

On the otherhand discrete Walsh transformation offers new methods for real-time signal processing and digital hardware implementation. There exist fast transform algorithms and no multiplications are required in Walsh transforms except sign changes. The correlation is not completely removed in the transform domain because there are non-vanishing elements outside the diagonal of the following transform

$$\frac{1}{N} \underline{W} \text{Cov}(\underline{x}) \underline{W} = \underline{\Delta}$$

If the covariance matrix is a dyadic convolution-type matrix the process is a wide-sense dyadic-stationary process and the correlation of the process \underline{x} is eliminated by linear Walsh transformation [9].

Because there are not many non-vanishing elements outside the diagonal we derive our motivation for an application of the Walsh transform.

The discrete Walsh transform of the finite intervals with the length $N=2^n$ is defined as

$$\hat{x}(\nu) = \underline{W} x(\nu)$$

with the inverse transform

$$x(\nu) = \frac{1}{N} \underline{W}^T \hat{x}(\nu)$$

\underline{W} is the N -th order Walsh-matrix and the elements $w(j,k)$ take only the values $+1$ and -1 . The matrix is orthonormal [10,11]

$$\underline{W} = \underline{W}^T = N \underline{W}^{-1}$$

Data compression is achieved when the variances in the spectral domain of the spectral signal to be processed are known. This can be done by computer simulation [12]. According to the mean energy distribution G. Robinson has derived the bit-assignment for the spectral coefficients in a system with $N=16$ and proved good results.

For the real-time signal processing with high order systems ($N=16 - 1024$) as required for example in image processing we developed a transform processor based on Walsh functions, other two-valued and also three-valued functions. Analog hardware implementations have already been discussed [13,14]. But using modern integrated MSI and LSI circuits the system may be implemented with minimized expense and highest flexibility. In the following chapter the implementation is discussed in detail.

Digital Hardware Implementation

Figure 3. shows the basic diagram of a transform processor.

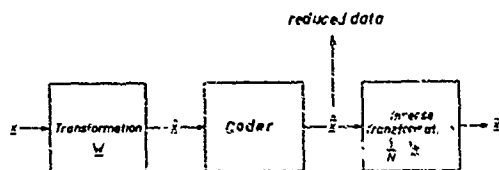


Figure 3. Basic diagram of a transform-processor.

The subsystem called spectral coder can also be replaced by other units for linear and non-linear filtering. With a logarithmic unit the system allows the double spectrum analysis in the Walsh domain.

The transform and inverse transform subsystems are shown in figure 4.

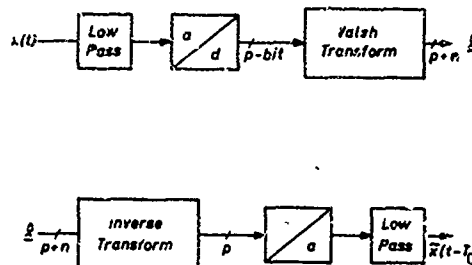


Figure 4. Block diagram of the transform units

Any arbitrary waveform is frequency limited in a low-pass frequency filter with respect to the sampling theorem. The discrete values at times kat are converted into binary words with finite length p . The quantized signal has here an input quantization error. The transform of the binary input signal yields N spectral coefficients. If the word-length is $p+n$ no additional errors are made in the transformation process. The original binary signal word is given by the inverse transform which is the same as the transform. The binary word is shifted to the right and thus multiplied with $1/N$. The d/a converter is producing an analog signal which is the same as the quantized input signal except a time delay of Nat or N samples. A low pass filtering yields a time continuous frequency limited signal.

The total number of required operations in a matrix multiplication is N^2 . There exist fast algorithms called Fast Walsh Transform (FWT) reducing the required operations to $N \log N$ [15,16]. A transform onto sequence order spectra requires additional storage and the complete input vector $x(\nu)$. A transform onto sequence order however will be useful for comparing the results from the well known Fourier spectrum direct with the Walsh spectrum because there is a first order one-to-one correspondence between sequence and frequency [17]. Therefore and because of more flexibility an ordinary matrix multiplication processor has been worked out instead of a fast transformer.

The matrix multiplication is computed in N^2 steps and the transform vector is given by N partial subsums

$$\hat{x}(\nu) = \underline{W} x(\nu) = [w_0, w_1, \dots, w_{N-1}] x(\nu)$$

where w_k are the columns of the Walsh matrix

$$\hat{x}(\nu) = x_0 w_0 + x_1 w_1 + \dots + x_k w_k + \dots + x_{N-1} w_{N-1}$$

x_0, x_1, \dots, x_{N-1} are the elements of the ν -th signal vector.

A k -th partial sum of the transformation vector is given by

$$\hat{x}_k(\nu) = [w_0, w_1, \dots, w_k] x(\nu)_k$$

$$\begin{pmatrix} \hat{x}_0 \\ \hat{x}_1 \\ \vdots \\ \hat{x}_j \\ \vdots \\ \hat{x}_{N-1} \end{pmatrix}_k = \begin{pmatrix} w_{0,0} & \dots & w_{0,k} \\ w_{1,0} & \dots & w_{1,k} \\ \vdots & & \vdots \\ w_{j,0} & \dots & w_{j,k} \\ \vdots & & \vdots \\ w_{N-1,0} & \dots & w_{N-1,k} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_k \\ \vdots \\ x_{N-1} \end{pmatrix}_\nu$$

The Matrix operations consist of N^2 sign changes and additions. N operations are computed serially in each sampling interval Δt . With parallel binary arithmetic the clock frequency of the machine is given by

$$f_{\text{clock}} = N f_{\text{sampl.}} = \frac{N}{\Delta t}$$

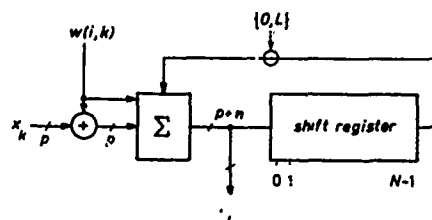
The logic (TTL and MOS) is limiting the clock to f_{cmax} so that the upper signal band limit is

$$f_{\text{up}} = \frac{f_{\text{cmax}}}{2N}$$

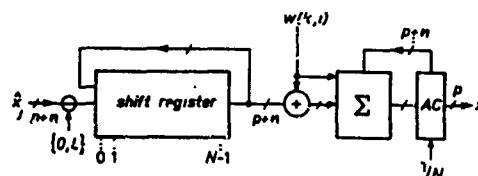
The implemented processor has a clock frequency of 2 MHz and the upper signal band limit is 15.6 kHz at $N=64$. There is no limit for low frequencies and long intervals.

Figure 5 shows the block diagram of the transform and inverse transform units. For the binary representation we use the two's complement because there exists a simple rule for the multiplication with +1 and -1. The arithmetic unit contains parallel exor-gates adders and shiftregisters. We regard now the state k at the time $k\Delta t$. The binary input word is multiplied N -times with the column vector w_k of the Walsh matrix. At $w_k = -1 \equiv L$ the word is inverted and an L is added to the last significant bit. The N products are added to the N foregoing partial sums which are stored in the N -stage shift-register. The new sums \hat{x}_k are read into the first stage of the shift register and shifted cyclically to the right with the

clock frequency. After $N(N-1)$ operations the input word x_{N-1} is multiplied with the last column of the Walsh matrix, added to the sums $\hat{x}_{(N-2)}$ and at the output of the adder the Walsh coefficients appear serially. In the feedback between shiftregister and adder there are AND-gates for the multiplication with zero. This is required at the beginning of each interval and also useful for the multiplication with the three-valued functions $\{-1, 0, +1\}$.



Transform



Inverse transform

Figure 5. Block diagram of the arithmetic units of the processor

During the last sampling interval $(N-1)\Delta t$ the coefficients $\hat{x}_0, \hat{x}_1, \dots$ appearing at the output of the transform unit are read into the shift register of the inverse transform unit. This register has an intern recirculation logic. In each sampling interval the register content is shifted cyclically to the right and at the output of the last stage the N Walsh-coefficients will appear. The multiplication with the corresponding row of the Walsh matrix and summation yield the original signal word x_k . The word is now shifted n -times to the right according to the multiplication with $1/N$. Because of the symmetry of the Walsh-matrix the signs for transform and inverse transform are the same and only one Walsh-generator is needed. We have used a proposal of Peterson [18].

The arithmetic units as shown above can be also used as digital Walsh-filters simply modifying the register commands [19, 20].

Spectral Coding

The first investigations with the transform processor have shown energy distributions as shown in figure 6.

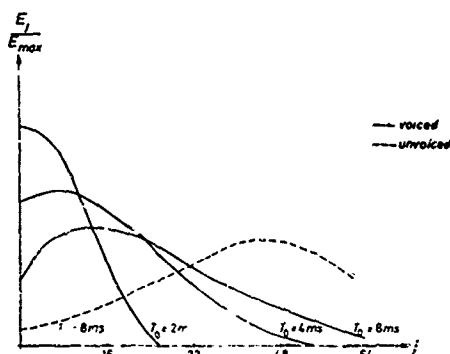


Figure 6. Qualitative energy distribution of German speech in the Walsh-spectrum

Because the absolute spectral distribution is constant, the short time spectrum is differing very much at changes of the transform intervals from short duration as 2ms to long one as 8ms. But the total energy of the signal is always constant in the spectral domain because Parseval's theorem holds true; the areas under the curves are the same. The distribution will also be different for voiced and unvoiced parts. The distribution of figure 6. is nearly the same as in the Fourier case. The voiced vowels have their maximum at the absolute sequences $\phi = i/T_0 \approx 1\text{kHz}$ (kilo zero crossings per sec.) which is equivalent to 1kHz in the frequency spectrum. The unvoiced parts have their maximum at higher sequences (3-6 kHz) according to 3-6kHz in the Fourier spectrum.

For $N=64$ and $T_0=8\text{ms}$ the energy distribution of the voiced parts has been quantized as shown in figure 7.

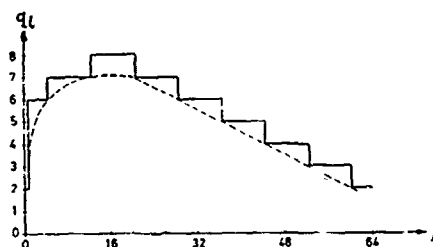


Figure 7. Bit Assignment in the Walsh-spectrum

q_i is the number of bits for the quantization of each coefficient. The quotient between input bit-rate (8-bit PCM) and spectrum rate thus is given as reduction factor of

$$r = \frac{64\text{kb/s}}{44\text{kb/s}} = 1,455$$

and there was no difference and loss in quality between input and output signal. Experiments have also been done with higher reduction up to 22kb/s and there was no loss in intelligibility and only a small amount of increasing quantization noise especially in the unvoiced parts. But this is also clear because the distribution of the voiced parts was taken. Further improvement will be achieved if voiced and unvoiced parts are coded with different schemes.

Pitch-synchronous Walsh Transform

High data compression however is only possible if the speech significant characteristics are turned out better. Thus, the bit-rate of the voiced quasiperiodic vowels can be reduced far-reaching using pitch-synchronous adaptive Walsh transformation. The main problem is the pitch detection, which must be determined exactly for reducing the variances in the several spectra. Figure 8. shows the basic diagram of an orthogonal Walsh-vocoder.

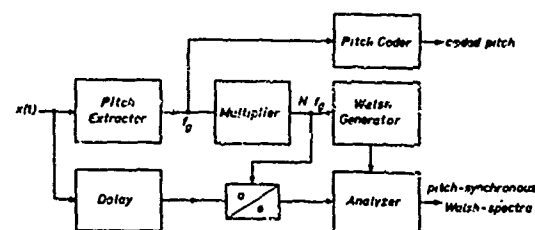


Figure 8. Basic diagram of an orthogonal adaptive Walsh-vocoder

The pitch frequency is multiplied by N and the delayed speech signal is sampled adaptive so that each period consists of N samples. The sample frequency is also controlling the Walsh generator and the arithmetic unit. Pitch and coefficients could be coded effectively with differential PCM. The estimated bit-rate might be 1-2kb/s. Though other systems allow higher data compression [21] the advantage of the

Walsh transformation lies in the minimal elaborateness. A photograph of a prototype transform processor is shown in figure 9.



Figure 9. Walsh transform processor

Acknowledgement

The work was sponsored by the Deutsche Bundespost, FTZ-FI Darmstadt.

References

- [1] Gallager, R.G. Information Theory and Reliable Communication, J. Wiley & Sons, Inc. New York, London, Sydney, 1968
- [2] Musmann, H.G. Über Lineare Transformationen zur Redundanzreduktion, Nachrichtentechn. Fachberichte, Band 40, 1971 Codierung, VDE-Verlag GmbH. Berlin 12 ed. by W. Endres
- [3] Flanagan, J.L. Speech Analysis, Synthesis and Perception, Springer New York 1965
- [4] Gethöffer, H.G. Sequence Analysis using Correlation and Convolution, Proc. of the 1971 Symp. on Appl. of Walsh Functions, Naval Research Lab. Washington D.C. AD 727000, pp 118
- [5] Klein, W. The Transformation errors of the Walsh Vocoder, Nachrichtent. Zeitschrift NTZ 23 (1970) pp. 126-128
- [6] Davenport, W.B., Root, W.L., Random Signals and Noise, McGraw-Hill, New York, Toronto, London, 1958 pp. 96-101
- [7] Musmann, H.G. Über Redundanzreduzierende Quellencodierung, Habilitationsschrift, TU Braunschweig 1970
- [8] G. Robinson, Orthogonal Transform Feasibility Study, Monthly Progress Report 6 Dec. 1970 Comsat Laboratories, Clarksburg/Maryland
- [9] Pearl, J. Walsh Processing of Random Signals, Proc. of the 1971 Symp. on Appl. of Walsh Functions, Naval Research Lab. Washington, D.C. AD 727 000
- [10] Harmuth, H., Transmission of Information by Orthogonal Functions, Springer Berlin, Heidelberg, New York 1969
- [11] Walsh, J.L. A Closed Set of Orthogonal Functions, American Journal of Mathematics Vol. 55 1923, pp. 5-24
- [12] Campanella, S.J., Robinson G., A Comparison of Walsh and Fourier Transformations for Application to Speech, Proc. of the 1971 Symp. on Appl. of Walsh Functions, Trans. Electro. Magn. Comp. EMC-13 (71) 3, pp 199-201
- [13] Böttwetter, C. Analog Sequence Analysis and Synthesis of Voice Signals, Proc. of the 1970 Symp. on Appl. of Walsh Functions, Naval Research Lab. Washington, D.C. pp. 220-229
- [14] Vandiver, E.F. A Flexible Walsh Filter Design for Signals for Moderately Low Sequence, Proc. of the 1970 Symp. on Appl. of Walsh Functions, Naval Research Lab. Washington, D.C. pp. 3-6
- [15] Whelchel, J.E., Guinn, D.F., The Fast Fourier Hadamard Transform and its Use in Signal Classification and Representation, Aerospace Electronics Conference EASCON record 1968 pp. 561-571
- [16] Kremer, H. Algorithmen für die Schnelle Walsh Transformation, ANT-Forschungsbericht 12 Dec. 1970
- [17] Böttwetter, C., Gethöffer, H. Die Gegenseitige Spektraldarstellung von Walsh Funktionen und Trigonometrischen Funktionen, Nachrichtentechnische Zeitschrift NTZ 24 (1971) 4 pp. 189
- [18] Peterson, H.L. Generation of Walsh Functions, Proc. of the 1970 Symp. on Appl. of Walsh Functions, Naval Research Lab. Washington D.C. pp. 55-57
- [19] Gethöffer H. Convolution and Deconvolution with Walsh Functions, Proc. of the 1971 Symp. on Theory and Applications of Walsh Functions, The Hatfield Polytechnic, Hatfield Hertf. UK
- [20] Gethöffer, H. Mutual Mapping of Generalized Convolution Systems, Proc. of the 1972 Symp. on Applications of Walsh Functions
- [21] Atal, B.S., Hanauer S.L. Speech Analysis and Synthesis by Linear Prediction of the Speech wave, J.A.S.A. 50(2) 1972 pp. 637-655

WORD RECOGNITION BY MEANS OF WALSH TRANSFORMS

by

Moyett T. Clark, Dept. of Defense, Ft. Meade, Md.
John E. Swanson, Dept. of Defense, Ft. Meade, Md.
John A. Sanders, Dept. of Defense, Ft. Meade, Md.

Abstract

An experiment is described in which word recognition is based on comparing the Walsh sequence components of spoken words with those of a stored library of words. A Walsh transform computer program is used to calculate the sequence components of successive segments of spoken words. The components for each word to be recognized are arranged into an amplitude - sequence - time matrix and correlated against a set of known test matrices. A test matrix is generated for each word by averaging six matrices of the word. By using the highest correlation coefficient to determine which word is spoken, recognition scores between 89.9% and 100% are achieved for a ten word vocabulary spoken by four speakers.

Introduction

In the recognition technique described in this paper, amplitude - sequence - time matrices for ten spoken words are prepared for four speakers, and these matrices are compared by means of a decision rule.

A Walsh transform computer program calculates the sequence components for successive sections of each spoken word. After the discrete Walsh sequence components of a spoken word are determined, a computer program arranges the components into a matrix of identifying numbers. By correlating the matrices with those of a stored library of words, the spoken word is identified by the largest correlation coefficient.

The objective of the pilot experiment is to recognize a limited vocabulary of words and to take advantage of the speed of the Walsh transform which requires only real time additions and subtractions.

*Harmuth [1] has defined sequence to be one half the average number of zero crossings per second of a function and has abbreviated sequence as "zps" in analogy to cycles per second.

Walsh Spectral Analysis of Speech

If a segment of a speech signal $f(t)$ is represented by a sequence of N samples, $f(nT)$, $0 \leq n \leq N-1$, the discrete Walsh Fourier or Hadamard transform can be defined as [2]

$$W(k) = \sum_{n=0}^{N-1} f(nT) \text{wal}(k,n), k=0,1,\dots,N-1 \quad (1)$$

where T is the sampling interval. Similarly, the inverse Walsh transform is

$$f(nT) = \frac{1}{N} \sum_{k=0}^{N-1} W(k) \text{wal}(n,k), n=0,1,\dots,N-1 \quad (2)$$

where N is an integral power of two.

The first two discrete Walsh functions are defined as

$$\text{wal}(0,n)=1, \text{ for } n=0,1,2,\dots,N-1 \quad (3)$$

$$\text{wal}(1,n) = \begin{cases} 1, & \text{for } n=0,1,2,\dots,N/2-1 \\ -1, & \text{for } n=N/2, N/2+1,\dots,N-1 \end{cases} \quad (4)$$

The remainder of the set of Walsh functions can be generated by the following iterative equation:

$$\text{wal}(k,n) = \text{wal}([k/2], 2n) \text{wal}(k-2[k/2], n)$$

where $[k/2]$ indicates the integer part of $k/2$.

In this paper the discrete Walsh transform of $f(nT)$, $0 \leq n \leq N-1$ is computed on an IBM 360/85 by an algorithm developed by Ulman [3]. The algorithm provides the components $W(k)$, $0 \leq k \leq N-1$, in order of sequence.

If the even and odd discrete Walsh components in (1) are combined, a power spectrum can be computed as follows [1,4]:

$$P(k) = \begin{cases} W^2(0), & k=0 \\ W^2(2k) + W^2(2k+1), & k=1,2,\dots,N/2-1 \end{cases} \quad (5)$$

The computations in (1) and (5) provide only one spectral section, that is, the sequence components at a time $t=(N-1)T$. To obtain a short-time spectral analysis, we compute (1) and (5) at successive instants of time and essentially determine a running Walsh spectrum for each spoken word.

Hence

$$W_r(k) = \sum_{n=0}^{N-1} f(nT+rMT) \text{wal}(k,n), k=0, \dots, N-1 \quad (6)$$

and

$$S_r(k) = \begin{cases} W_r^2(0), & k=0 \\ W_r^2(2k) + W_r^2(2k-1), & k=1, 2, \dots, N/2-1 \end{cases} \quad (7)$$

The two sets of numbers $W_r(k)$ and $S_r(k)$ represent the discrete Walsh transform and power spectrum, respectively, of a section of a speech signal starting at $t=rMT$ and ending at $t=rMT+(N-1)T$. Successive sections are spaced in time by MT [5].

A normalized spectrum, $Q_r(k)$, is calculated from (7) for each word by finding the maximum value of $S_r(k)$ for each utterance and modifying (7) as

$$Q_r(k) = -10 \log \frac{S_r(k)}{S_{r,\max}}, k=0, 1, \dots, N/2-1 \quad (8)$$

All values of $Q_r(k)$ below -30db are set equal to -30db. Hence

$$P_r(k) = \begin{cases} -30\text{db} & \text{if } Q_r(k) \leq -30\text{db} \\ Q_r(k) & \text{if } Q_r(k) > -30\text{db} \end{cases} \quad (9)$$

Each word is represented by an amplitude - sequency - time matrix whose values are between 0 and -30db.

General Procedure

The spoken digits, one through ten are low-pass filtered to 4kHz, sampled at 10kHz by a 12 bit PCM coder, and recorded on a digital tape. The digital tape is edited so as to produce six records, each containing 1624 sample data points, or $6 \times 1024 = 6144$ samples for each spoken word.

The utterances on the digital tape are processed by the Walsh spectrum analyzer described by (6) to (9). The sets of numbers $P_r(k)$ for successive sections of each word are arranged in the computer as a data matrix with the format shown in Figure 1. The columns of the matrix correspond to successive segments of a word, where each segment starts at time $t=rMT$ and ends at time $t=rMT+(n-1)T$. The rows are the sequency components $P_r(k)$, $0 \leq k \leq N/2-1$, which correspond to each word segment. Such a matrix may be regarded as a digital spectrogram [5,6].

For each utterance, a test matrix is generated by averaging six matrices of the same word. Thus there are ten test matrices for each speaker. The words to be recognized for a particular speaker are compared with each of the speaker's test matrices.

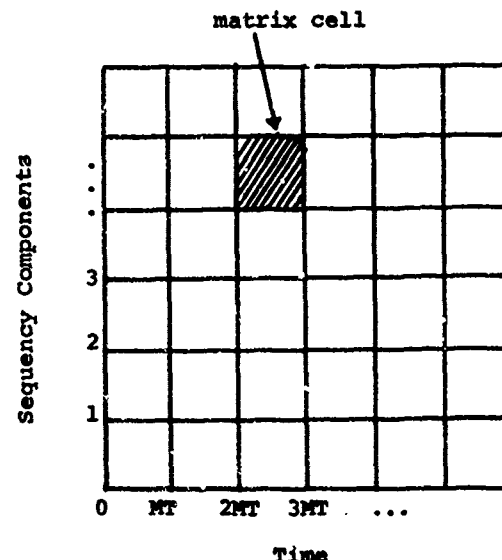


Figure 1. Example of the coordinates of the amplitude - sequency - time matrix

Let $C(i,j)$ represent a matrix cell in a word to be recognized, where i is the index for the sequency components and j is the index for successive time sections. A correlation coefficient can be generated for the two matrices as follows [7]:

$$\rho = \frac{\sum_{i=1}^{N/2} \sum_{j=1}^L C(i,j) D(i,j)}{\sqrt{\sum_{i=1}^{N/2} \sum_{j=1}^L C^2(i,j)} \sqrt{\sum_{i=1}^{N/2} \sum_{j=1}^L D^2(i,j)}} \quad (10)$$

$N/2$ is the number of components in the Walsh power spectrum and L is the total number of successive time segments. By generating a correlation coefficient between the unknown word and each of the test matrices, a decision can be made on the basis of the best correlation coefficient.

Experimental Procedure

Three experiments were performed. For the first and second experiments, the parameter N in (6) to (10) was chosen as 128. This means that separation between successive spectral samples was 78.125 zps (zps is one-half the average number of zero

	Recognition Scores				Approximate Computer Time	
	First male speaker	Second male speaker	Third male speaker	Female speaker	Generation of spectral components	Correlation
First Experiment No. of sequency components= $N/2=64$ No. of spectral sections= $L=48$	96.6%	94.9%	100.0%	93.3%	80 sec	34 sec
Seco ' Experiment Lower order components used No. of sequency components= $N/4=32$ No. of spectral sections= $L=48$	93.3%	93.3%	96.6%	91.6%	80 sec	21 sec
Third Experiment No. of sequency components= $N/2=32$ No. of spectral sections= $L=96$	96.6%	94.9%	100.0%	89.9%	112 sec	42 sec

Table 1. Recognition scores for four speakers and approximate computer time required to generate, correlate, and recognize words.

crossings per second [1]). The parameter M was chosen as 128 in (6) to (9). With the sampling interval $T=6.1$ ms, $M=128$ corresponds to obtaining spectral sections every $MT=12.8$ ms. Since each utterance consists of 6144 sample data points, there were 48 spectral sections for each spoken word. Hence L in (10) was 48.

In the first experiment the successive spectral sections, $P_r(k)$, $0 \leq k \leq N/2-1$, contained 64 components. Each word to be recognized was represented by a 64×48 amplitude - sequency - time matrix and correlated against each of the 64×48 test matrices.

In the second experiment only the lower order sequency components, $P_r(k)$, $0 \leq k \leq 31$, were used. Each word was represented by a 32×48 matrix and correlated against each of the 32×48 test matrices. The results of the first and second experiments are shown in Table 1.

For the third experiment the parameter N in (6) to (10) was chosen as 64, and the separation between sequency samples was 156.25 zps. By choosing $M=64$, we obtained 96 spectral sections, where the sections were separated by $MT=6.4$ ms. The successive spectral sections, $P_r(k)$, $0 \leq k \leq N/2-1$, contained 32 components. Each word was represented by a 32×96 matrix and correlated against each of the 32×96 test matrices. The results of the third experiment are also shown in Table 1.

Experimental Results

Twelve sets of the numbers one to ten were spoken by three male and one female speaker. The first six

sets were used to form the test matrices. Matrices for the second six sets were compared against the test sets. The recognition rates which varied from 89.9% to 100% are shown in Table 1. Table 1 also shows the approximate computer time necessary to generate and correlate the spectral components for each experiment.

In the second experiment an attempt was made to reduce the computer computation time and storage requirement by using the lower order sequency components. Since the results do not vary significantly from those of the first experiment, it appears that we were able to achieve data reduction. This is similar to the results obtained by Pratt, Kane, and Andrews [8] who were able to achieve bandwidth reduction by ignoring the higher order sequency components.

In experiment number three the sequency components were further apart. The recognition scores of the first and third experiments were similar except for the female speaker, but more computer time was required for the third experiment. Therefore, the first experiment produced satisfactory overall results.

Discussion

Because the Walsh transform requires less computer time than the Fourier transform, the Walsh transform was chosen for the recognition experiments. The Walsh transform requires only real additions and subtractions whereas the Fourier transform requires complex multiplications, additions, and subtractions. Pratt, Kane, and Andrews [8] were able to realize a reduction in time by a

factor of 6.6 by using the Walsh transform.

Hence the objective of this experiment was to recognize words with a minimum amount of computer time and storage by correlating Walsh spectrums. It required approximately 80 to 112 seconds of computer time on an IBM 360/85 to generate matrices for twelve sets of 10 spoken words. It also required approximately 21 to 42 seconds to generate the test matrices and perform the recognition procedure.

The major disadvantage of the system was its inability to cope with the variations of the duration of the spectral events which constitute words. This problem was partially solved by using test matrices which were the average of six matrices.

References

- 1) H.F. Harmuth, "A Generalized Concept of Frequency and Some Applications," IEEE Transactions on Information Theory, Vol. IT-14, May, 1968, pp. 375-382
- 2) John L. Shanks, "Computation of the Fast Walsh - Fourier Transform," IEEE Transactions on Computers, Vol. C-18, May 1969, pp. 457-459
- 3) H.L. Ullman, "Computation of the Hadamard Transform and the R - Transform in Ordered Form," IEEE Transactions on Computers (Correspondence), April 1970, pp. 359-360
- 4) S.J. Campanella and G.S. Robinson, "Digital Sequence Decomposition of Voice Signals," Proceedings of Applications of Walsh Functions Symposium Workshop, March 31 - April 3, 1970, Naval Research Lab, Washington, D.C.
- 5) A.V. Oppenheim, "Speech Spectrograms Using the Fast Fourier Transform," IEEE Spectrum, August 1970, pp. 57-62
- 6) M.H.L. Hecker, "Speech Recognition an Interpretive Survey of the Literature," ASHA Monographs No. 16, American Speech and Hearing Association, Washington, D.C., January, 1971.
- 7) M.T. Clark, "Word Recognition by Means of Orthogonal Functions," IEEE Transactions, Vol. AU-18, Sept. 1970, pp. 304-312.
- 8) W.K. Pratt, J. Kane, and H.C. Andrews, "Hadamard Transform Image Coding," Proceedings of IEEE, Vol. 57, January, 1969, pp. 56-68.

A PROGRAMMABLE WALSH FUNCTION GENERATOR AND ITS USE IN A HIGH SPEED INVERSE TRANSFORM APPARATUS

Dr. A. R. Elliott Mrs. S. Mikhail
Department of Electrical Engineering and
The Communications Research Laboratory of
McMaster University,
Hamilton, Ontario, Canada.

Abstract

A novel circuit for generating the first 64 Walsh functions with no timing error, hazard-free, and synchronising pulses for the time period θ is explained. The circuit requires a minimum amount of hardware, and can be operated at high speeds. Its use in an inverse Walsh Transform apparatus for displaying video data is covered.

Introduction

The project outlined here was undertaken when an interest was shown in using Walsh functions for image transfection by some authors (1,2). The possibility of transmitting the image in the transform domain, and then reconstructing it at the receiver offered the possibility of a lower bit-rate transfer of data, and hence the possibility of bandwidth reduction. In order to use this transform economically, inexpensive hardware had to be developed which would work at the high speeds demanded by video processing for real-time applications. The results outlined below are the result of an initial investigation into such hardware.

Walsh Generator Design

The typical Walsh function generator uses exclusive-or (3,4,5) or exclusive-nor gates (6), to produce the appropriate Walsh functions. Generally these are based on combinations of Radamacher functions. One design requires the use of discrete resistor-capacitor components (7) to produce the final function output. A recent circuit (8) has the advantage of being able to determine the exact value of the Walsh function at any point in the time interval θ . Except for the circuit of (8) these hardware generators produce a variable time delay in generating the higher frequency functions, since the generation of the higher frequency terms depends on the generation of one or more of the lower frequency functions. When attempting to reproduce time varying functions from their Walsh domain coefficients, this timing error may be critical, particularly at high frequencies.

Present techniques also have a tendency to produce so-called "hazards", or stray pulses, simply because of the inherent techniques used in generating the functions. These hazards are most notable when generating functions using exclusive-or gates and previously generated terms. In order to produce a better circuit for generating these functions, the problems of hazards, accurate timing of the zero-crossing

during the period θ , and exact knowledge of the beginning of the time period θ , along with high frequency operation, and a simple circuit are required. The circuit proposed here accomplishes all these aims, and is also programmable. The input of a standard binary code for the Walsh number N (between 0 and 64) will determine which function is generated. Because of the relationship between the SAL and CAL functions, and the Walsh functions (WAL) the circuit can also generate these.

The selected Walsh function is generated for the entire period θ . A "SYNC" pulse is produced at the beginning or end of the time period θ . The circuit is shown in Fig. 1.

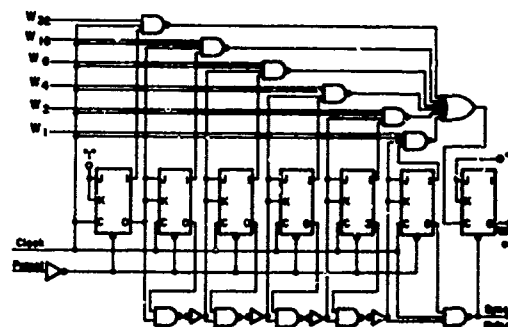


Fig. 1: A 6-bit Programmable Walsh Function Generator.

The Walsh numbers are entered using the weighting shown on the W input lines. For example, WAL(9,0) would be obtained by entering the following binary code:

W_{32}	W_{16}	W_8	W_4	W_2	W_1
0	0	1	0	0	1

This binary code is gated by AND gates coupled to a 6-bit binary down counter, and the clock input. By using a narrow clock pulse, the circuit can produce a pulse train depending on the state of the counter. In effect, the zero-crossings of the Radamacher functions are detected and the appropriate ones are used to toggle a final flip-flop. Because the output is taken from a memory element (the final flip-flop), its

* Patent applied for, January, 1970.

output state is held at a fixed level until the next clock pulse is permitted to toggle the memory. Accordingly, transitions within the circuit do not appear at the output, and the "hazard" problem is overcome. Because the output is tied to the clock frequency, changes of the output always occur at "clock" times, and hence there is no rippling through of the final answer. The circuit was also designed to be fully iterative (i.e. each stage in the main circuit is identical, allowing for a simple addition or deletion of stages depending on the number of Walsh series required). Iterative design, however, does cut down on the speed of operation of the circuit. If a fully synchronous circuit were used for the counter, very high speed could be built. With standard logic modules (series 54 or 74 TTL logic), the circuit would work up to 10MHz.

The circuit also generates a "SYNC" pulse at the beginning of the time period θ which is effectively used in the high speed transform apparatus described later to transfer data from a buffer register (coupled to a computer or manual switches) to the register holding the coefficient number $WAL(N, \theta)$. Hence the data presented on the W lines of Fig. 1 is held for the entire time period θ , and is only updated at the beginning of the period. The buffer register can be easily loaded while the main generator is producing the actual functions.

Series Expansion by Walsh Functions

Any periodic function $f(t)$ can be expanded in a series of the orthogonal system of Walsh functions $WAL(N, t)$ in the interval of orthogonality T . By making $\theta = t/T$, the system will be normalized, and the expansion of $f(\theta)$ will be given as:

$$f(\theta) = \sum_{N=0}^{\infty} A_N WAL(N, \theta) \quad (1)$$

$$\text{for } 0 \leq \theta < 1$$

The coefficients A_N of the series expansion can be obtained by multiplying equation (1) by $WAL(j, \theta)$, and integrating the products over the period of orthogonality using the orthogonality relationship:

$$\int_0^1 WAL(N, \theta) WAL(j, \theta) d\theta = \delta_{Nj} \quad (2)$$

$$\text{where } \delta_{Nj} = 1 \text{ for } N = j$$

$$\delta_{Nj} = 0 \text{ for } N \neq j$$

Hence the i^{th} coefficient is given by

$$A_i = \int_0^1 f(\theta) WAL(i, \theta) d\theta \quad (3)$$

The series expansion of $f(\theta)$ can be expressed in terms of the even and odd Walsh functions $CAL(i, \theta)$ and $SAL(i, \theta)$ as

$$f(\theta) = a_0 + \sum_{i=1}^{\infty} [a_i CAL(i, \theta) + b_i SAL(i, \theta)] \quad (4)$$

where

$$a_0 = \int_0^1 f(\theta) d\theta$$

$$a_i = \int_0^1 f(\theta) CAL(i, \theta) d\theta$$

$$b_i = \int_0^1 f(\theta) SAL(i, \theta) d\theta$$

The first few Walsh functions and their relation to the SAL and CAL functions are shown in Fig. 2.

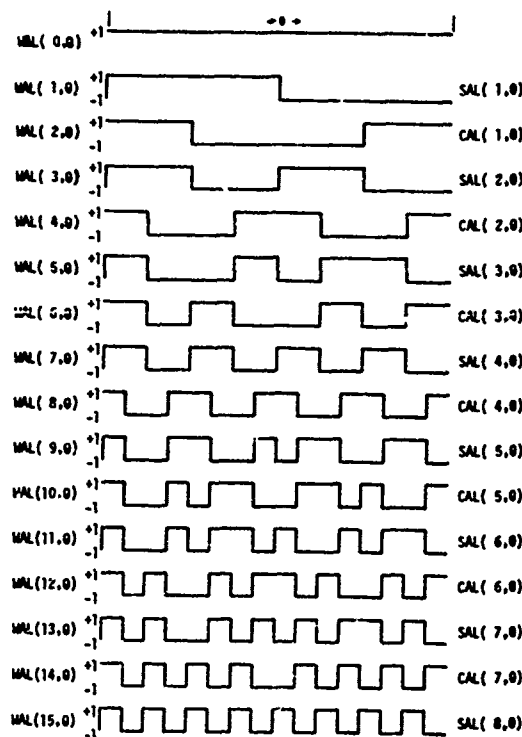


Fig. 2: The first 16 Walsh Functions.

Amplitude Sampling and Walsh-Fourier Analysis

The sampling theorem of Fourier analysis states that a signal band limited to B Hz is completely determined by $2B$ amplitude samples per second. This theory is also true for Walsh-Fourier analysis.

Let a signal $f(t)$, band limited to B Hz., be represented by amplitude samples taken at a rate equal to $2B$ samples per second. If the period for the time base is taken as T such that $2BT$ is equal to $2^N = N$ (the number of samples of the signal $f(t)$), and making $\theta = t/T$, then the expansion of the signal $f(t)$ in a normalized Walsh series will be:

$$f(\theta) = \sum_{i=0}^{\infty} a_i WAL(i, \theta) \quad 0 \leq \theta < 1 \quad (5)$$

The coefficients will be given by equation (3).

From the skew symmetry property of Walsh functions, the Walsh-Fourier coefficients a_i for $i \geq N$ will vanish. This means that the maximum frequency contained in the signal $f(t)$ in the interval T is $(2BT-1)$. This will be the upper limit in the summation of equation (5), and the discrete transform becomes:

$$f(\theta) = \sum_{i=0}^{N-1} a_i \text{WAL}(i, \theta) \quad (6)$$

In other words, the maximum frequency of the signal $f(t)$ is B zero crossings per second. By representing $f(\theta)$ by N samples in the interval $0 \leq \theta < 1$, the integration of equation (3) for determining the coefficients can be changed to a summation by dividing the period θ into N sub-intervals of length $1/N$. Then the coefficients are given by:

$$a_i = \frac{1}{N} \sum_{\theta=0}^{N-1} f(\theta) \text{WAL}(i, \theta) \quad (7)$$

$$i = 0, 1, \dots, N-1.$$

These equations can be implemented to perform the Fast Walsh Transform, and the coefficients a_i obtained.

The Instrument Design

The fundamental limitation of the Walsh Function generator described earlier to be programmable to obtain the first 64 coefficients limits the number of samples N to 64. The instrument was designed to operate with dominant term synthesis, and a selection of the dominant 16 terms from the field of 64 is used in the final design.

The instrument was designed to perform the inverse transform (effectively equation (6)) but only using dominant terms. The main objective of the circuit was to directly drive a television screen to display video data being generated in the Walsh domain. The long term objective of the work was to investigate bandwidth reduction schemes by transforming an original image to the Walsh domain, transmitting significant terms, and then reconstructing the image at the receiver.

The general circuit for the hardware implementation is shown in Fig. 3, where only eight terms are shown being generated. The circuit is duplicated and summed in the final adder in order to obtain the 16 term synthesis described. The final circuitry was analog in nature in order to maintain speeds around 1 to 10 MHz, and used IC analog multipliers to perform the arithmetic. The coefficients a_i are converted to analog form by D/A converters, and the output of the Walsh function generator directly drives the other terminal of the multiplier. The output of each multiplier is, in effect, an amplitude modulated Walsh function. The outputs of the multiplier are added using a standard operational amplifier adder, and the final output drives a CRT intensity grid.

A synchronizing pulse is used to lock the horizontal sweep generator to the standard video frame sync. Fig. 4 shows the test circuit apparatus, with a manually preset signal being

generated continuously across the screen. The time period θ was taken as $1/4$ of the screen trace time.

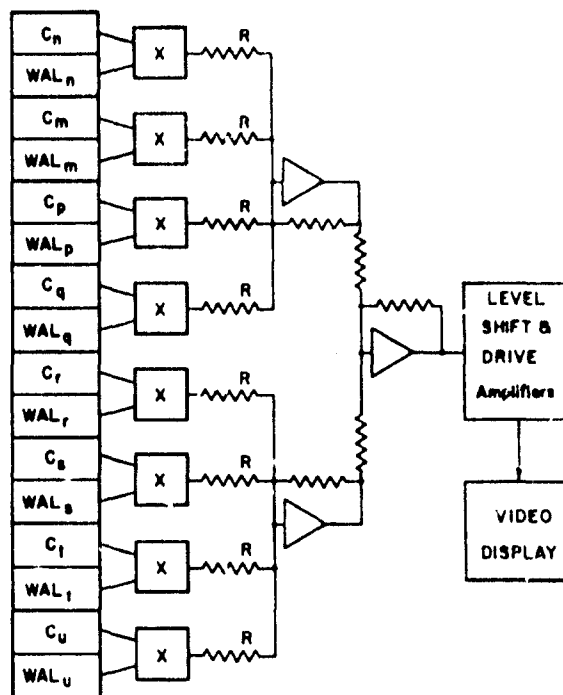


Fig. 3: Inverse Transform Basic Circuit.

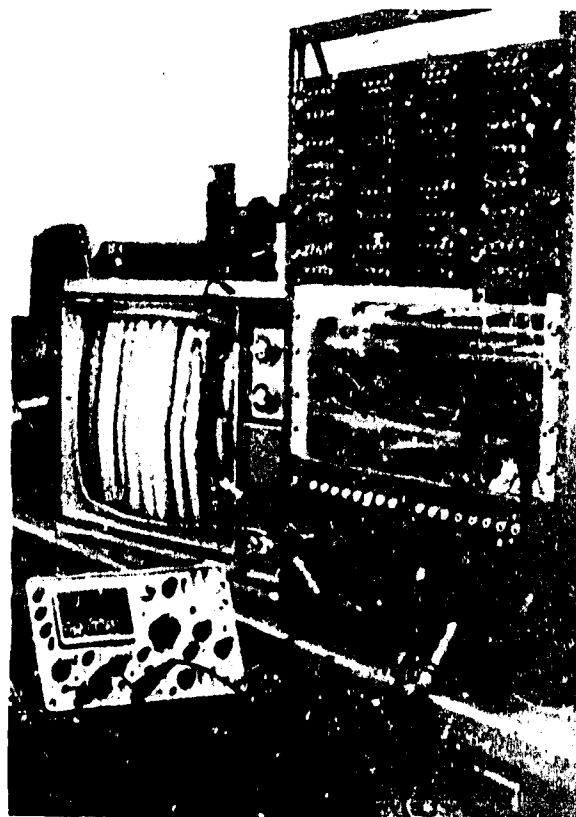


Fig. 4: Test Apparatus and Inverse Transform Instrument.

Results and Conclusions

The system has been constructed and tested for manual input of data. Line traces of a typical television display have been analysed to obtain the Walsh coefficients a_n , and these have been implemented on the instrument in order to measure errors.

Some tentative results on using more bits to specify the dominant coefficients, and fewer bits to represent the smaller coefficients, as well as dominant terms selection have shown that bit-rates of 2.2 to 3 bits/coefficient, and using 64 coefficients to represent a line (four updates of 16 coefficients each) produce a reasonable reconstruction.

Further work is in progress in dividing the line into 8 intervals, and interfacing the instrument to a computer, for direct control of the video display.

The instrument is designed to reconstruct video signals, based on dominant term synthesis, and hence the need for a programmable Walsh function generator that will work at these high speeds.

References

1. W.K. Pratt, J. Kane, H.C. Andrews, "Hadamard Transform Image Coding", Proc. IEEE, Vol. 57, No. 1, pp.56-58, Jan., 1969.
2. T.S. Huang, W.F. Schreiber, O.J. Tretiak, "Image Processing", Proc. IEEE, Vol. 59, No. 11, Nov. 1971.
3. Harnuth, H., Transmission of Information by Orthogonal Functions, Springer-Verlag, Berlin/New York, 1969.
4. I.A. Davidson, "The Use of Walsh Functions for Multiplexing Signals", Proc. Symp. Appl. Walsh Functions, Washington, 1970, pp. 23-25.
5. K.H. Siemens, R. Kitai, "Digital Walsh-Fourier Analysis of Periodic Waveforms", IEEE Trans. on Instrumentation and Measurements, IM-18, No. 4, pp. 316-321, Dec. 1969.
6. S.H. Manoli, "Walsh Function Generator", Proc. IEEE, Vol. 59, No. 1, pp. 93-94, Jan. 1971.
7. F.J. Lebert, "Walsh Function Generator for a Million Different Functions", Proc. Symp. Appl. Walsh Functions, Washington, 1970, pp. 52-54.
8. R. Kitai, K. Siemens, "A Hazard-Free Walsh Function Generator", Trans. Inst. and Measurements, pp. 80-83, Feb. 1972.

COMPUTATION OF THE FAST HADAMARD TRANSFORM

Y.Y. Shum A.R. Elliott
Department of Electrical Engineering and
the Communications Research Laboratory
McMaster University
Hamilton, Ontario, Canada

Abstract

The sequence of +1's and -1's of any Hadamard function can be deduced from its binary index. The factorization of a Hadamard matrix is derived similarly. An "in place" Fortran subroutine to compute the Hadamard transform of a real signal is developed.

Hadamard Matrices

The Hadamard matrix is a square array of plus and minus ones, whose rows and columns are orthogonal to each other (i.e., the product of the matrix and its transpose is the identity matrix times a constant N). N is the order of the matrix, and the lowest value it may assume is two, thus giving the lowest ordered Hadamard matrix as

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (1)$$

Hadamard matrices of higher order, for N a power of two, may be generated by a Kronecker product operation, such that

$$H_{2N} = \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix} \quad (2)$$

For example,

$$H_4 = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \quad (3)$$

$$H_8 = \begin{bmatrix} H_4 & H_4 \\ H_4 & -H_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \quad (4)$$

Generation of Hadamard Functions

Each row of a Hadamard matrix corresponds to a Hadamard function $had(j,k)$, for $j = 0, 1, 2, \dots, N-1$. It is well known that these functions may be generated through the first m Rademacher functions [1]. For example, with $N = 2^3$, the values of the Rademacher functions, namely r_0, r_1 and r_2 , for $k=0, 1, 2, \dots, 7$, are as illustrated in Fig. 1.

k	0	1	2	3	4	5	6	7
r_0	1	1	1	1	-	-	-	-
r_1	1	1	-	-	1	1	-	-
r_2	1	-	1	-	1	-	1	-

Fig. 1: Rademacher Functions

For this example, j can be expressed in a binary notation as

$$j = 4b_4 + 2b_2 + b_1 \quad (5)$$

for $b_i = 0$ or 1 ,

$$i = 1, 2, 4.$$

Fig. 2 shows all the possible values of j .

j	b_4	b_2	b_1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Fig. 2: Binary Equivalent of j

The correct combination of r_0, r_1 and r_2 , to produce $had(j,k)$ is given by the binary 1's under columns b_4, b_2 and b_1 respectively, as shown in Fig. 2. For $b_i = 0$, the corresponding Rademacher function is not involved as a multiplying factor and, in its place, for all values of k , a value of 1 is assumed.

From the binary equivalent of j , the entire

sequence of +1's and -1's for any Hadamard function $\text{had}(j,k)$ can be deduced [2]. The initial value of this sequence is +1 as $\text{had}(j,0)$ is equal to 1. The length of the sequence is determined by the number of binary digits in j such that, for every b_i , the sequence is to be extended to the right by i bits. The binary nature of this system means that this extension can be achieved simply by doubling its original length. Accordingly, the bits to be examined successively are $b_1, b_2, \dots, b_i, \dots, b_{N/2}$.

As mentioned earlier, if $b_i = 0$, a multiplying factor of +1 is assumed. Consequently, the original string of +1's and -1's is to be copied once more to the right. On the other hand, if $b_i = 1$, the corresponding Rademacher function is involved in the product. Its value is equal to +1 for $k = 0, 1, \dots, i-1$, but -1 for $k = i, i+1, \dots, 2i-1$. Hence the sequence is to be expanded by adding on the complement of the existing string of +1's and -1's. The formulation of $\text{had}(j,k)$, where $j = 101$ in binary form, is illustrated in Fig. 3. This simple technique to generate any Hadamard function through its binary indexing is valid as long as the length of the sequence is a power of two.

Hadamard function $\text{had}(j,k)$ of length $N=2^m$
Let $j = b_1 + 2b_2 + \dots + ib_i + \dots$,

where $b_i = 0$ or 1,

$$i = 1, 2, 4, \dots, N/2.$$

For every b_i , extend the sequence of +1's and -1's to twice its length by adding the sequence on the right under the following rules:

- 1) the same sequence if $b_i = 0$,
- 2) the complement of the sequence if $b_i = 1$.

For example, with $N=8$,

$$j = b_1 + 2b_2 + 4b_4$$

$$\text{had}(5,k) = \text{had}(101,k)$$

Sequence

1	initial value
1 -	$b_1 = 1$
1 - 1 -	$b_2 = 0$
1 - 1 - - 1 - 1	$b_4 = 1$

Fig. 3: Generation of a Hadamard Function Through Binary Indexing

Hadamard Transform of a Real Signal

The Hadamard transform of a real signal may be defined as

$$F(j) = \frac{1}{N} \sum_{k=0}^{N-1} f(k) \cdot \text{had}(j,k) \quad (6)$$

where $F(j) = j^{\text{th}}$ normalized Hadamard coefficient,

$f(k)$ = discrete samples of the signal,

$\text{had}(j,k) = j^{\text{th}}$ Hadamard function.

The inverse Hadamard transform is given by

$$f(k) = \sum_{j=0}^{N-1} F(j) \cdot \text{had}(k,j) \quad (7)$$

Since the Hadamard matrix is orthogonal,

$$\text{had}(j,k) = \text{had}(k,j) \quad (8)$$

Equations (6) and (7) are analogous, except for the factor N . Consequently the procedure to compute the Hadamard transform and the inverse process are basically the same.

Equation (6) may be expressed in an iterative form, thereby improving computational efficiency. For simplicity of explanation, consider the case when $N=8$. Then j and k may be represented by the following binary equivalents:

$$j = 4j_4 + 2j_2 + j_1 \quad (9)$$

$$k = 4k_4 + 2k_2 + k_1 \quad (10)$$

where $j_i, k_i = 0$ or 1.

The Hadamard function may be completely factorized [3,4] into

$$\text{had}(j,k) = (-1)^{j_1 k_1} (-1)^{j_2 k_2} (-1)^{j_4 k_4} \quad (11)$$

Equation (6) may now be rewritten in an iterative form [3,4]

$$F(j_4, j_2, j_1) = \frac{1}{8} \sum_{k_1=0}^1 (-1)^{j_1 k_1} \sum_{k_2=0}^1 (-1)^{j_2 k_2} \sum_{k_4=0}^1 (-1)^{k_4 j_4} f(k_4, k_2, k_1) \quad (12)$$

The array from $\sum_{k_4=0}^1 (-1)^{k_4 j_4} f(k_4, k_2, k_1)$ becomes

the data array for the next stage of computation each of which requires $N/2$ additions and $N/2$ subtractions. Eventually the Hadamard coefficient

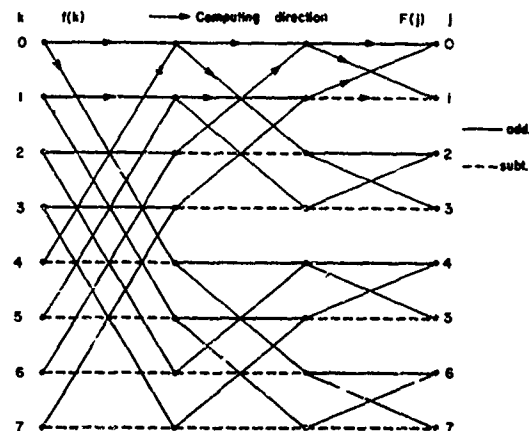


Fig. 4: Signal Flow Graph for Fast Hadamard Transform

icients will be produced in their correct order without any shuffling [4]. The signal flow graph corresponding to equation (12) is shown in Fig. 4.

Computation of the Fast Hadamard Transform

The fast Hadamard transform may also be derived from the binary notation of j , where j is the index of the Hadamard coefficient. With $N=8$, Fig. 2 and Fig. 4 can be referenced again. As will be shown, the locations (j, b_i) under columns b_4, b_2, b_1 in Fig. 2, are related to the corresponding nodes on the signal flow graph of Fig. 4.

From the graph, it may be apparent that the terms (any term appearing in the flow graph will be given the general symbol d) concurring at any node differ in their indexing by i . The same result can be obtained by noting where asymmetry occurs under column b_i , as indicated by the horizontal lines in Fig. 2. The mathematical operator associated with these two terms is determined by $(-1)^{j, b_i}$ or $[1-2x(j, b_i)]$. If $(j, b_i) = 0$, the two terms d_j and d_{j+i} are summed. On the other hand, if $(j, b_i) = 1$, a subtraction should take place. From Fig. 4, the correct formulation is $d_{j-i} - d_j$.

Any two terms that are summed are also involved in a subsequent subtraction, and the results can be stored in the same memory locations, such that

$$d_j = d_j + d_k \quad (13)$$

$$d_k = d_j - d_k \quad (14)$$

where d_j, d_k = sampled data, intermediate results or Hadamard coefficients,

$$k = j + 2^i$$

$$i = 0, 1, 2, \dots, m-1,$$

$$j = 0, 1, 2, \dots, N-1, \text{ excluding those of } k.$$

For a general-purpose computer, computations are done in a serial fashion. Consequently, "in place" arithmetic requires the modification of equation (14) to

$$d_k = d_j - d_k - d_k \quad (15)$$

Computation of the inverse Hadamard transform can be formulated from equations (13) and (15).

In computing the Hadamard transform, the normalization factor of $1/N$ ($N=2^m$) may be incorporated into each of the m stages of computation as a factor of $1/2$, thereby preventing the possibility of an "overflow" in integer arithmetic. Accordingly, the basic equations for the Hadamard transform are

$$d_j = 1/2(d_j + d_k) \quad (16)$$

$$d_k = d_j - d_k \quad (17)$$

For example, with $i = 4$:

$$d_r = 1/2(d_0 + d_4), d_4 = d_0 - d_4$$

$$d_1 = 1/2(d_1 + d_5), d_5 = d_1 - d_5$$

$$d_2 = 1/2(d_2 + d_6), d_6 = d_2 - d_6$$

$$d_3 = 1/2(d_3 + d_7), d_7 = d_3 - d_7$$

The next two stages of computation, for $i=2,1$, may be similarly developed. The general steps to implement the fast Hadamard transform based on binary notation correspond to the signal flow graph of Fig. 4, except for the necessary modification mentioned earlier.

```
SUBROUTINE FPHT2 (NR,N,M)
  DIMENSION MR(N)
  C HADAMARD TRANSFORM OF A REAL SIGNAL
  C MR = SAMPLED DATA
  C N = NO. OF DATA
  C 2**M = NO. OF COEFF. PER TIME INTERVAL
  L = N
  K = 1
  DO 3 NM = 1, M
    I = 0
    L = L/2
    DO 1 NK = 1, K
      DO 2 NL = 1, L
        I = I + 1
        J = I + K
        4 MR(I) = (MR(I)+MR(J))/2
        1 MR(J) = MR(I)-MR(J)
        2 I = J
        3 K = K*2
      RETURN
    END
```

For the inverse Hadamard transform, the lines with the same statement numbers are to be replaced:

```
4 MR(I) = MR(I) + MR(J)
1 MR(J) = MR(I) - MR(J) - MR(J)
```

Conclusions

The entire sequence of 1's and -1's of a Hadamard function can be deduced from its binary indexing. A fast Hadamard transform algorithm, using "in place" computations, is similarly developed. The same algorithm can be modified to compute the inverse transform.

References

- [1] K.W. Henderson, "Some Notes on the Walsh Functions", IEEE Trans. Computers, Vol. C-13, pp. 50-52, February, 1964.
- [2] D.A. Swick, "Walsh Function Generation", IEEE Trans. Information Theory, Vol. IT-15, No. 1, p. 167, January, 1969.
- [3] J.L. Shanks, "Computation of the Fast Walsh-Fourier Transform", IEEE Trans. Computers, Vol. C-18, pp. 457-459, May, 1969.

[4] K.W. Henderson, "Comment on 'Computation of the Fast Walsh-Fourier Transform'", IEEE Trans. Computers, Vol. C-19, p. 850, September, 1970.

[5] L.J. Ulman, "Computation of the Hadamard Transform and the R-Transform in Ordered Form", IEEE Trans. Computers, Vol. C-19, pp. 359-360, April, 1970.

A PAPALLEL ARRAY HARDWARE IMPLEMENTATION OF THE FAST HADAMARD AND WALSH TRANSFORMS

A.R. Elliott Y.Y. Shum
Department of Electrical Engineering and
the Communications Research Laboratory
McMaster University
Hamilton, Ontario, Canada

Abstract

The design of a relatively simple, but very fast, digital circuit that can generate both the Hadamard or Walsh coefficients is explained. A slight modification to the circuit allows the inverse transform to be performed as well. A useful communication application is described.

Introduction

A previous paper [1] has described an algorithm for implementing the fast Hadamard transform of an N-length data array, where $N = 2^m$, using "in place" computation. A hardware implementation of the algorithm illustrated by Fig. 4 of [1] requires m adders for each coefficient when working in a parallel mode of operation. If $m = 1$, the hardware cost per coefficient is minimized. This can be achieved by recycling the output from the adders into the same storage registers m times. The loss in speed with such an arrangement is immaterial for most practical applications.

Factorization of the Hadamard Matrix

It is well known that an efficient way to implement the Hadamard transform of an N-length ($N=2^m$) real signal is to decompose the transform matrix into m factors that have many zero elements, thereby reducing the number of arithmetic operations. For example,

$$H_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \end{bmatrix} \quad (1)$$

$$= G_0 \cdot G_1 \cdot G_2 \quad (2)$$

$$= P \cdot P \cdot P \quad (3)$$

$$= Q \cdot Q \cdot Q \quad (4)$$

$$\text{where } G_0 = \begin{bmatrix} 1 & 1 & . & . & . & . & . & . \\ 1 & - & . & . & . & . & . & . \\ . & . & 1 & 1 & . & . & . & . \\ . & . & . & . & 1 & 1 & . & . \\ . & . & . & . & . & . & 1 & - \\ . & . & . & . & . & . & 1 & 1 \\ . & . & . & . & . & . & 1 & - \\ . & . & . & . & . & . & 1 & - \end{bmatrix} \quad (5)$$

$$G_1 = \begin{bmatrix} 1 & . & 1 & . & . & . & . & . \\ . & 1 & . & 1 & . & . & . & . \\ 1 & . & - & . & . & . & . & . \\ . & 1 & . & - & . & . & . & . \\ . & . & . & . & 1 & . & 1 & . \\ . & . & . & . & . & 1 & . & 1 \\ . & . & . & . & . & . & 1 & - \\ . & . & . & . & . & . & . & 1 \end{bmatrix} \quad (6)$$

$$G_2 = \begin{bmatrix} 1 & . & . & . & 1 & . & . & . \\ . & 1 & . & . & . & 1 & . & . \\ . & . & 1 & . & . & . & 1 & . \\ . & . & . & 1 & . & . & . & 1 \\ 1 & . & . & . & - & . & . & . \\ . & 1 & . & . & . & - & . & . \\ . & . & 1 & . & . & . & - & . \\ . & . & . & 1 & . & . & . & - \end{bmatrix} \quad (7)$$

$$P = \begin{bmatrix} 1 & 1 & . & . & . & . & . & . \\ . & . & 1 & 1 & . & . & . & . \\ . & . & . & . & 1 & 1 & . & . \\ . & . & . & . & . & . & 1 & 1 \\ 1 & - & . & . & . & . & . & . \\ . & . & 1 & - & . & . & . & . \\ . & . & . & . & 1 & - & . & . \\ . & . & . & . & . & . & 1 & - \end{bmatrix} \quad (8)$$

$$Q = \begin{bmatrix} 1 & . & . & . & 1 & . & . & . \\ 1 & . & . & . & - & . & . & . \\ . & 1 & . & . & . & 1 & . & . \\ . & . & 1 & . & . & . & - & . \\ . & . & . & 1 & . & . & . & 1 \\ . & . & . & . & 1 & . & . & . \\ . & . & . & . & . & 1 & . & 1 \\ . & . & . & . & . & . & 1 & - \end{bmatrix} \quad (9)$$

Each factor G_i , where $i = 0, 1, 2$, is an orthogonal matrix. Multiplication of G_i and a column matrix requires only additions and subtractions and the arithmetic may be done "in place". A fast Hadamard transform algorithm has been implemented [1] based on the concept of equation (2).

Matrix Q is the transpose of P, or vice versa. Consequently the characteristics of both are analogous. It may be sufficient to describe the properties of P, and deduce those of Q on similar guidelines.

Computation of the Fast Hadamard Transform

It may be observed from equation (8) that the product of matrix P and a one-dimensional array involves merely additions and subtractions of data that are adjacent, and the storing of the results in memory registers that are $N/2$

locations apart. In the typical computer, mathematical calculations are done in a serial fashion. Hence, in software simulation, at least one other array is required as a buffer, as illustrated by the array LR in the Fortran subroutine PHTR2 shown below. Each factor P corresponds to one stage of computation. After m stages, the Hadamard coefficients are located in the original data array.

```

SUBROUTINE PHTR2 (JR,LR,N,M)
  DIMENSION JR(N),NR(N)
  C  HADAMARD TRANSFORM OF A REAL SIGNAL
  C  JR = DATA ARRAY
  C  LR = BUFFER ARRAY
  C  N = 2**M ; NO. OF DATA
  NH = N/2
  LF = 1
  M1 = AND(M,1)
  IF (M1.NE.0) GO TO 20
  NM = N
  K = NH
  GO TO 22
20  IN = M-1
  K = NH/2
22  LL = K
30  JF = LF+1
  IN = 1
  DO 4 I = 1,NM
    J = JF
    IF (IN.LT.0) GO TO 2
    DO 1 L = LF,LL
      5  LR(L) = (JR(J-1)+JR(J))/2
      6  LR(L+K) = LR(L)-JR(J)
      1  J = J+2
      GO TO 4
    2  DO 3 L = LF,LL
      7  JR(L) = (LR(J-1)+LR(J))/2
      8  JR(L+K) = JR(L)-LR(J)
      3  J = J+2
    4  IN = -IN
    IF (M1.EQ.0) RETURN
    IF (M1.LT.0) GO TO 40
    M1 = -1
    LF = LF+NH
    LL = LL+NH
    GO TO 30
40  DO 9 I = 1,NH
    J = I+NH
    10  JR(I) = (JR(I)+JR(J))/2
    9  JR(J) = JR(I)-JR(J)
  RETURN
END

```

To perform the inverse Hadamard transform, the lines with the same statement numbers are to be replaced by the following:

```

5  LR(L) = JR(J-1)+JR(J)
6  LR(L+K) = JR(J-1)-JR(J)
7  JR(L) = LR(J-1)+LR(J)
8  JR(L+K) = LR(J-1)-LR(J)
10 JR(I) = JR(I)+JR(J)
9  JR(J) = JR(I)-JR(J)-JR(J)

```

Hardware Implementation of the Fast Hadamard and Walsh Transforms

In a parallel-operating system, the same memory elements may be utilized to hold the original sampled data, intermediate results, and eventually, after m operations, the required

Hadamard coefficients in their correct order. A preliminary design shows that it is possible to use complement arithmetic along with M.S.I. adder packages available, and D-type flip-flops arranged as registers for the sampled data as well as the coefficient values.

A general rule for recycling of the results from the output of the adders, based on equation (8), is thus given by:

$$d_j = \frac{1}{2}(d_k + d_{k+1}) \quad (10)$$

$$d_{j+N/2} = \frac{1}{2}(d_k - d_{k+1}) \quad (11)$$

where $j = 0, 1, 2, \dots, (N/2-1)$,

for $k = 0, 2, 4, \dots, N-2$.

Fig. 1 demonstrates a proposed digital circuit that will implement the fast Hadamard transform for the determination of 8 coefficients. The expansion to a circuit that will obtain the first 64 coefficients is straightforward and will have the same general structure as the circuit of Fig. 1. Under the assumption that a 7-bit coefficient value is sufficient, with the eighth bit handling transient arithmetic overflow, the cost per coefficient, with off-the-shelf components, would be approximately \$12.00.

For $N = 2^6$, (i.e., 64 coefficients) the circuit need only be clocked 6 times, with adequate time between clock pulses to allow the output of the adders to settle. With modern IC's this should be accomplished in less than 200 nanoseconds. A circuit for obtaining 128 coefficients would take about 250 nanoseconds.

The Hadamard coefficients may be rearranged to obtain the Walsh ordering. In a hardware implementation to obtain the Walsh coefficients, a straightforward approach would be to add a new set of registers at the output of the adders to hold the Walsh coefficients. This is illustrated in Fig. 1.

Note that the forward and the inverse transforms differ only by a simple factor of 2[1]. Accordingly, a shifting of the bit lines returning to the storage registers by one bit will allow the inverse transform to be performed by the same hardware.

Application of the Parallel Hardware Array

A possible communication system using this novel hardware design is shown in Fig. 2. This is a combination transmitter-receiver assumed to be used in a mobile communication system (e.g. an aircraft) in such a way that the bit-rate of transmission is lower than standard PCM systems. Others [3,4,5] have demonstrated the feasibility of this approach. The circuit proposed here is a tentative design for a speech communications system. It determines and transmits the eight dominant coefficients out of a field of 64 in either the Hadamard or Walsh domains. Similarly, it can receive

"dominant" coefficients and reconstruct the necessary waveform. Consequently, for such a communication system the relevant data are formatted as the amplitude of the coefficient and the corresponding coefficient number. These data words are transmitted over standard data links to similar transmitter-receivers. It is estimated that a total package of this type would cost about \$3,000 - \$5,000. Future research will be aimed at constructing such a package and comparing its operation with other forms of digital transmission of speech.

Conclusion

It is shown that the decomposition of a Hadamard matrix into identical factors leads to the implementation of a feasible software subroutine and a high-speed hardware design for the fast Hadamard transform. The same hardware can be modified for the fast Walsh transform, and for the inverse transform as well. A tentative design of a digital transmitter-receiver system is proposed.

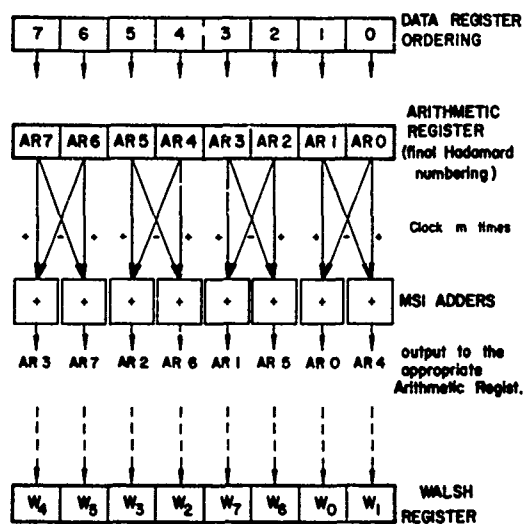


Fig. 1: Typical Parallel Array for 8 Hadamard or Walsh Coefficients

References

- [1] Y.Y. Shum and A.R. Elliott, "Computation of the Fast Hadamard Transform", Proc. Symp. Appl. of Walsh Functions, Washington, 1972.
- [2] H.C. Andrews and K.L. Caspari, "A Generalized Technique for Spectral Analysis", IEEE Trans. Computers, Vol. C-19, No. 1, pp. 16-25, Jan. 1970.
- [3] C. Boßwetter, "Analog Sequence Analysis and Synthesis of Voice Signals", Proc. Symp. Appl. of Walsh Functions, Washington, 1970, pp. 220-229.
- [4] S.J. Campanella and G.S. Robinson, "Digital Sequence Decomposition of Voice Signals", Proc. Symp. Appl. of Walsh Functions, Washington, 1970, pp. 230-237.
- [5] Y.Y. Shum and A.R. Elliott, "Speech Analysis and Synthesis Using the Hadamard Transform", Proc. Symp. Appl. of Walsh Functions, Washington, 1972.

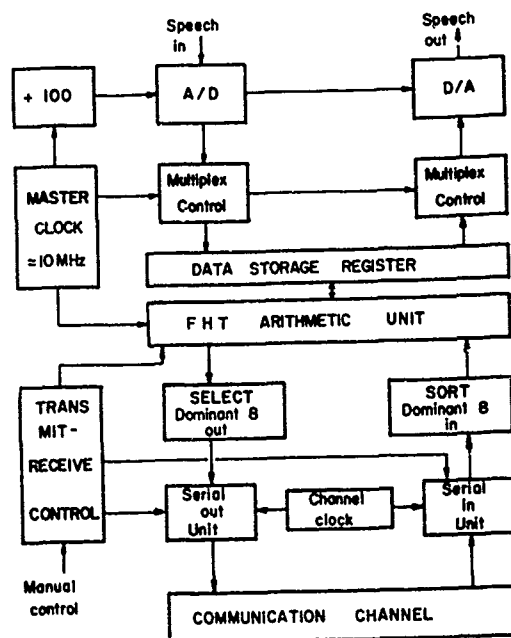


Fig. 2: Digital Transmitter-receiver System for Speech Communications

SPEECH ANALYSIS AND SYNTHESIS USING THE HADAMARD TRANSFORM

Y.Y. Shum A.R. Elliott
Department of Electrical Engineering and
the Communications Research Laboratory
McMaster University
Hamilton, Ontario, Canada

Abstract

An efficient algorithm is used to obtain 64 Hadamard coefficients for every 8 milliseconds of speech. Using only 8 or 4 of these coefficients, it is possible to reconstruct intelligible speech on a CDC-1700 computer system. The results of this investigation into speech analysis and synthesis in the Hadamard domain are presented.

Introduction

Several previous papers [1,2,3] have demonstrated the feasibility of using the discrete Hadamard or Walsh transform in image or speech processing with promising results. The immediate objective of this research is to evaluate the usefulness of transforming speech to the Hadamard domain, retaining several dominant coefficients per specific time interval, and then reconstructing the original signal from these coefficients. The long-range goal is to reduce the bit-rate necessary for speech communication over data links for those applications requiring intelligibility rather than high quality, and simultaneously to reduce the memory requirement of any associated hardware.

Computer Facility and Implementation

A block diagram of the CDC-1700 computer system and the apparatus used to perform this experiment is illustrated in Fig. 1. The present configuration of the system is such that the I/O bus can only handle A/D or D/A conversion one at a time, in a buffered mode. Consequently, real-time operation is not possible, although the algorithms used are sufficiently fast that real-time operation (actually a 16 msec. delay) can be contemplated in the future.

The original sentences for analysis and synthesis were taken from a master tape supplied by the 1972 International Conference on Speech Communication and Processing. The speech waveform was sampled at an 8KHz. rate, and the data were formatted into 14-bit words by the A/D converter. These digital samples were stored on a disk capable of holding 1.5 million words. Only 12 cylinders out of a total of 99 were used, which correspond to approximately 23 seconds of speech being handled on one pass. A fast Hadamard transform algorithm [4] was used to transform every group of 64 data words into the first 64 coefficients in the Hadamard domain. For the sampling rate used, the update time interval was thus 8 milliseconds. The dominant (in terms of largest absolute value) 8 (or 4)

coefficients were selected through another software algorithm. Synthesis was performed by using the dominant coefficients, their corresponding coefficient numbers, and the inverse Hadamard transform. The results were stored on the disk, destroying the original data file. A final D/A conversion through a low-pass filter produced the resultant analog signal that was recorded on tape.

Format and Bit-rate of Test Results

The tape recording presents the results as first, the "original" speech. This is simply the original digital samples sent directly through the D/A converter to demonstrate the degradation of the speech waveform due to the operating system. The bit-rate of data transfer is equivalent to 112,000 bits/seconds. Scaling the 14 bit words to 6 or 7 bits (about 48,000 bits/second) shows little change.

The second recording was speech reconstructed from the dominant eight coefficients of an analysed 64 coefficients, updated every 8 milliseconds. To specify the coefficient number, 6 bits are required. The coefficient amplitude values are of 7 bits (including sign). In effect, this corresponds to an average bit-rate of reconstruction of 13,000 bits/second.

The final recording was made using the four most dominant coefficients in the same 8 millisecond time interval. The reconstructed signal corresponds to a bit-rate of 6,500 bits/second.

The tentative results presented here are the beginning of a project to determine an effective digital scheme for minimizing the bit-rate of data transfer for intelligible speech communication. For this purpose, the algorithms used in the research are made sufficiently flexible such that changes may be made on the sampling rate, the update time interval, and the number of dominant coefficients for synthesizing speech. A possible outcome of this approach is the production of an inexpensive audio voice response unit for a computer that will require a minimum of memory storage. To this end, future research is aimed at analysing phonemes and diphones, with an attempt to create a digital vocabulary for voice synthesis.

Acknowledgements

The authors wish to thank Dr. S.S. Haykin, Director of the Communications Research Laboratory at McMaster University, who made computer facility available, and to C.M. Thorsteinson

for programming advice and assistance.

References

- [1] W.K. Pratt, J. Kane, and H.C. Andrews, "Hadamard Transform Image Coding", Proc. of the IEEE, Vol. 57, No. 1, pp. 58-68, January, 1969.
- [2] C. B88wetter, "Analog Sequency Analysis and Synthesis of Voice Signals", Proc. Symp. Appl. Walsh Functions, Washington, 1970, pp. 220-229.
- [3] S.J. Campanella and G.S. Robinson, "Digital Sequency Decomposition of Voice Signals", Proc. Symp. Appl. of Walsh Functions, Washington, 1970, pp. 230-237.
- [4] Y.Y. Shum and A.R. Elliott, "Computation of the Fast Hadamard Transform", Proc. Symp. Appl. of Walsh Functions, Washington, 1972.

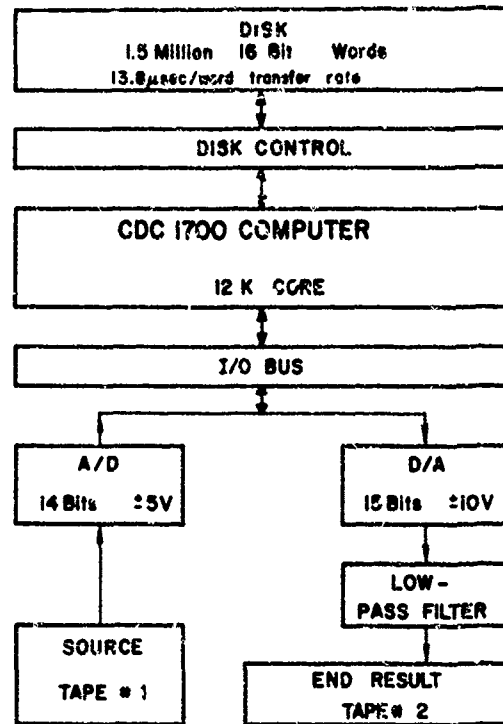


Fig. 1: CDC-1700 Computer System Installation at McMaster University

DIGITAL SIMULATION OF A LCS RESONANT FILTER FOR WALSH FUNCTIONS

Joe P. Golden
U.S. Naval Research Laboratory
Code 5712
Washington, D.C., U.S.A.

Introduction

An analog LCS serial resonant filter for Walsh functions is presented. Difference equations describing the lossless resonant filter were programmed on a general purpose computer to simulate the filter as a digital resonance filter [1-3]. The direct programming form for digital simulation was modified by using time variable coefficients to simulate the time variable switch necessary for tuning the filter to a discrete sequency. The steady state difference equations necessary for digital simulation of a lossy analog LCS resonant filter are also given.

Fundamentals

The serial resonance filter for Walsh functions as shown in Fig. 1 is basically the same as for sinusoidal functions except that it has a switch S which is used to tune the resonant filter. To illustrate the properties of this circuit, let switch S be closed, switch S_0 be open, $v_1(0) = V_1$, $v_2(0) = V_2$ and $C_1 = C_2 = C$. The current and voltages yield the following equations:

$$i(t) = \sqrt{\frac{C}{2L}} (V_1 - V_2) \sin \omega t \quad (1)$$

$$v_1(t) = V_1 - \frac{1}{2} (V_1 - V_2) (1 - \cos \omega t)$$

$$v_2(t) = V_2 + \frac{1}{2} (V_1 - V_2) (1 - \cos \omega t)$$

where $\omega = \sqrt{2/LC}$. At time $t = \pi/\omega$, the current goes to zero and the voltages are

$$v_1(\pi/\omega) = V_2 \quad (2)$$

$$v_2(\pi/\omega) = V_1$$

The important point is that the capacitor voltages at $t = 0$ and $t = \pi/\omega$ have interchanged values. To make use of this property, assume that the sampled Walsh function $\text{sal}(4, \theta)$ as shown in line 1 of Fig. 2 is applied to capacitor C_1 by instantaneously closing and opening

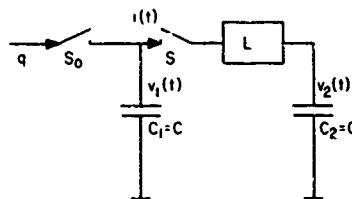


Fig. 1. Serial LCS Resonance Filter.

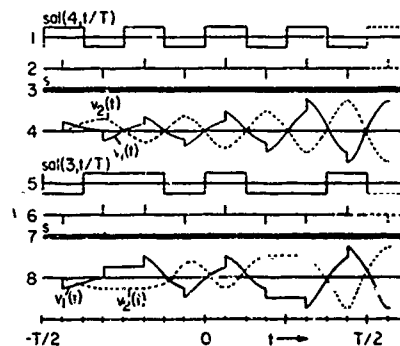


Fig. 2. Time Diagram for a LCS Resonance Filter Tuned to the Forcing Function.

switch S_0 . The sampled function $\text{sal}(4, \theta)$ must be in the form of a charge source if voltage is to be added accumulatively to C_1 . Let $v_1(t + T/2) = v_2(t + T/2) = 0$ and suppose $q = +V_1C$ is applied to C_1 at time $t = -7T/16$ as shown in line 2 of Fig. 2 which gives

$$v_1(t - 7T/16) = V_1 \quad (3)$$

$$v_2(t - 7T/16) = 0.$$

According to the results of equations (1) and (2), the voltages will interchange in the next time interval of $\Delta t = T/8$ which gives

$$v_1(t - 5T/16) = 0 \quad (4)$$

$$v_2(t - 5T/16) = V_1$$

as shown in line 4 of Fig. 2. The charge $q = -V_1C$ is then applied to C_1 at time $t = -5T/16$ according to the sampled Walsh function

sal(4,0). Again the voltages interchange and this building up process continues until steady state is reached. It should be noted that switch S is closed throughout this discussion, thus, tuning the filter to the forcing function sal(4,0). It is noted that the voltage $v_2(t)$ is the complement of the forcing function sal(4,0) which implies the desired condition for resonance.

Consider sal(3,0) as shown in line 5 of Fig. 2 with all initial conditions set to zero. The charge $q = -V_1C$ is applied to C_1 at time $t = -7T/16$ and the voltages are then allowed to interchange. The charge $q = +V_1C$ is applied at $t = -5T/16$, however, the voltages are not allowed to interchange this time. Switch S is opened at $t = -5T/16$ and it remains open for the duration $t = T/8$ as shown in line 8 of Fig. 2. The switch S could be opened at that time since the current is zero. At time $t = -3T/16$ switch S is closed and the charge $q = +V_1C$ is applied to C_1 . The voltages are allowed to interchange as before. If this process is continued with the switch S being opened (white \square) and closed (black \square) as shown in line 7 of Fig. 2, the voltages will build up with time which implies that the circuit is a serial resonant LCS filter for time sampled Walsh functions. The switching characteristics of switch S is the determining factor necessary for resonance to occur.

Fig. 3 shows what will happen if the forcing function sal(3,0) is applied to the circuit that is tuned to sal(4,0) and vice versa. The voltages do not build up with time which implies that the circuit is not tuned for resonance.

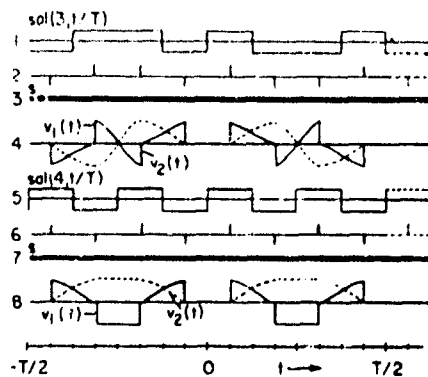


Fig. 3. Time Diagram for a LCS Resonance Filter Not Tuned to the Forcing Function.

The separation of the sum of two Walsh functions is illustrated in Fig. 4. If the input signal is sal(1,0) + sal(3,0) as shown on line 3 with the filter tuned to sal(3,0) as indicated by line 5, the output voltages v_1 and v_2 are

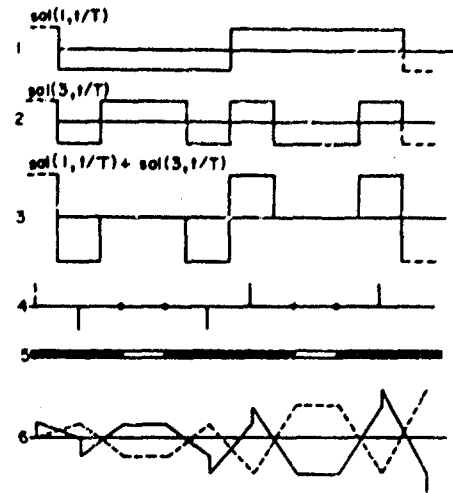


Fig. 4. Time Diagrams Illustrating Separation properties for LCS Resonance Filter for Sum of Two Walsh Functions.

shown in line 6 of Fig. 4. Note that v_2 clearly indicates the resonance condition.

Since the voltages $v_1(t)$ and $v_2(t)$ are only of interest at certain time points $t_n = n\pi/4\omega + T/16$ where $n = 0, \pm 1, \pm 2, \dots$ according to Fig. 2, difference equations may be used for the analysis of the circuit of Fig. 1. Denote $v_1(t_n)$ by $v_1(n)$ and $v_2(t_n)$ by $v_2(n)$ with switch S always closed and S_0 always open. According to equation (2) the voltages $v_1(n)$ and $v_2(n)$ are related by the following two difference equations of first order:

$$v_1(n+1) = v_2(n), \quad v_2(n+1) = v_1(n). \quad (5)$$

If the switch S_0 is momentarily closed at the time t_n , the applied charge $q_n = Cv_1'(n)$ increases the voltage across C_1 from $v_1(n)$ to $v_1(n) + v_1'(n)$. Equations (5) will assume the following form

$$v_1(n+1) = v_2(n), \quad v_2(n+1) = v_1(n) + v_1'(n). \quad (6)$$

Separation of the variables yields two difference equations of second order:

$$v_1(n+2) - v_1(n) = v_1'(n), \quad (7)$$

$$v_2(n+2) - v_2(n) = v_1'(n+1).$$

It would be instructive to carry the difference equation technique one step further by considering sal(3,0) as applied to Figs. 1 and 2 [4]. In analogy to (6) the following sets of equations are derived from Fig. 1 and line 8 of Fig. 2:

$$v_1(n+1) = v_2(n) \quad (8) \text{ (Cont.)}$$

$$v_1(n+2) = v_1(n+1) + v_1'(n+1) \quad (8)$$

(Cont.)

$$v_1(n+3) = v_2(n+2)$$

$$v_1(n+4) = v_2(n+3)$$

$$v_2(n+1) = v_1(n) + v_1'(n)$$

$$v_2(n+2) = v_2(n+1)$$

$$v_2(n+3) = v_1(n+2) + v_1'(n+2)$$

$$v_2(n+4) = v_2(n+3) + v_1'(n+3)$$

where $n = 0, 4, 8, 12, \dots, 4i, \dots$. Separation of the variables yields:

$$v_1(n+1) = v_1'(n-1) + v_1(n-1) \quad (9)$$

$$v_1(n+2) = v_1'(n+1) + v_1(n+1)$$

$$v_1(n+3) = v_2'(n) + v_1(n)$$

$$v_1(n+4) = v_1'(n+2) + v_1(n+2)$$

$$v_2(n+1) = v_1'(n) + v_2'(n-1)$$

$$v_2(n+2) = v_2(n+1)$$

$$v_2(n+3) = v_1'(n+1) + v_1'(n+2) + v_2(n)$$

$$v_2(n+4) = v_1'(n+3) + v_2(n+2)$$

where $n = 4i, i = 0, 1, 2, \dots$

Digital Simulation

The ideal digital LC resonant filter can be simulated on a general purpose digital computer by programming the difference equations which describe the particular resonance filter.

The simulation of the digital resonant filter tuned to $\text{sal}(4, \theta)$ can be accomplished by rearranging equations (7) by shifting back two time units which gives

$$v_1(n) = v_1'(n-2) + v_1(n-2) \quad (10)$$

$$v_2(n) = v_1'(n-1) + v_2(n-2).$$

Let the following changes in notation be made to aid in programming the variables $n \rightarrow J, v_1 \rightarrow V1, v_2 \rightarrow V2$ and $v_1' \rightarrow V1V$. Equation (10) now becomes

$$V1(J) = V1V(J-2) + V1(J-2) \quad (11)$$

$$V2(J) = V1V(J-1) + V2(J-2).$$

Equation (11) can be simulated in block diagram form using the direct form of programming

as shown in Fig. 5 where Δ indicates a time delay of one unit time. Figure 6 shows the results when the input forcing function $V1V = \text{sal}(4, \theta)$ with the filter tuned to $\text{sal}(4, \theta)$. Note that the filter resonant condition has been met as previously defined. If another Walsh function were applied as the input, resonance would not occur as simulated by Fig. 7 where $V1V = \text{sal}(3, \theta)$. The voltages $V1$ and $V2$ were limited to a \pm value. This limiting process is necessary since the digital simulation is unstable and the output voltages would continue to increase as long as the input is applied. This would, of course, be expected for a resonance circuit when applying an input at the same frequency as the filter is tuned. This is directly analogous to the resonance filters in the frequency domain.

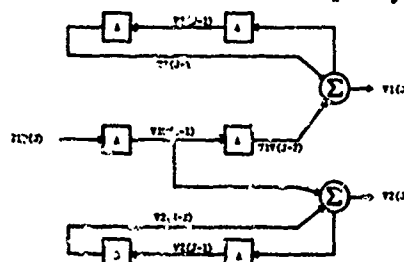


Fig. 5. Simulated Block Diagram of Digital Filter Resonance Tuned to $\text{sal}(4, \theta)$.

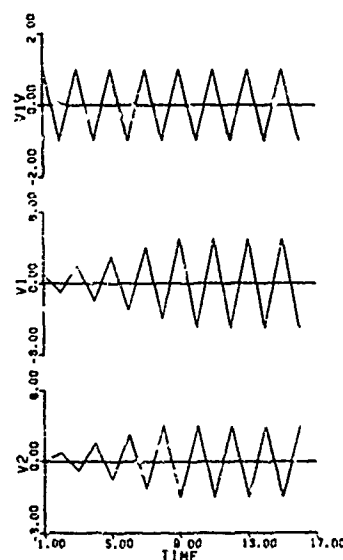


Fig. 6. Filter Tuned to $\text{sal}(4, \theta)$ with $V1V = \text{sal}(4, \theta)$.

The simulation of the digital filter shown in Fig. 5 which used the direct programming form had constant coefficients of either plus one or zero in the feedback and feed forward loops. The constant time invariant coefficients provided a rather simple simulation. Suppose

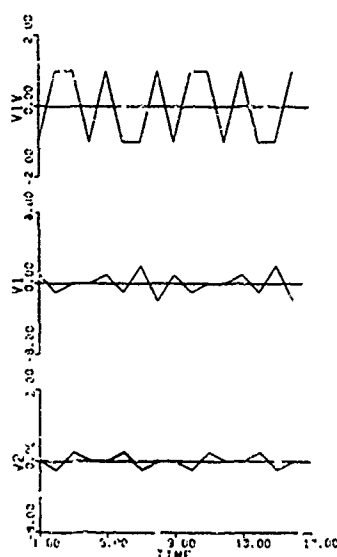


Fig. 7. Filter Tuned to $\text{sal}(4, \theta)$ with $V1V = \text{sal}(3, \theta)$.

the same type (direct) of programming form be used to simulate a digital filter tuned to $\text{sal}(3, \theta)$. This would require the coefficients to be time variable with values of plus one or zero depending on the time necessary to cause resonance. The simulated block diagram of the digital filter tuned to $\text{sal}(3, \theta)$ is shown in Fig. 8 where the Kronecker Delta is

$$\delta_{I+4, J} = 1 \quad \text{if } I + 4 = J \\ = 0 \quad \text{if } I + 4 \neq J.$$

Figure 8 is the simulation of equation (9) where the following notations have been changed to facilitate programming the equations $4i + 1 \rightarrow I + 4$, $4i \rightarrow I$, $m \rightarrow J$, $v_1 \rightarrow V1V$, $v_1 \rightarrow V1$, $v_2 \rightarrow V2$. Equation (9) can be rewritten using the Kronecker Delta notation as follows:

$$\delta_{I+4, J} \begin{bmatrix} V1(J) = V1V(J-2) + V1(J-2) \\ V2(J) = V1V(J-1) + V2(J-2) \end{bmatrix} \quad (12)$$

$$\delta_{I+5, J} \begin{bmatrix} V1(J) = V1V(J-1) + V2(J-1) \\ V2(J) = V2(J-1) \end{bmatrix} \quad (13)$$

$$\delta_{I+8, J} \begin{bmatrix} V1(J) = V1V(J-3) + V1(J-3) \\ V2(J) = V1V(J-2) + V1V(J-1) + V2(J-3) \end{bmatrix} \quad (14)$$

$$\delta_{I+7, J} \begin{bmatrix} V1(J) = V1(J-2) + V1(J-2) \\ V2(J) = V1V(J-1) + V2(J-2) \end{bmatrix} \quad (15)$$

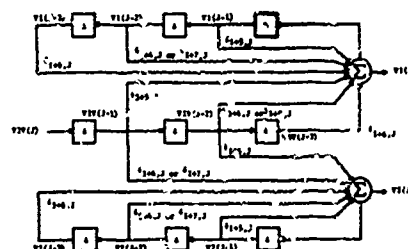


Fig. 8. Simulated Block Diagram of Digital Resonance Filter Tuned to $\text{sal}(3, \theta)$.

Figure 9 shows the digital simulation of the resonance filter tuned to $\text{sal}(3, \theta)$ with $\text{sal}(3, \theta)$ applied as the forcing function. The steady state condition is simulated the same as discussed earlier. The non-resonance condition is seen in Fig. 10 where the forcing function is $\text{sal}(4, \theta)$.

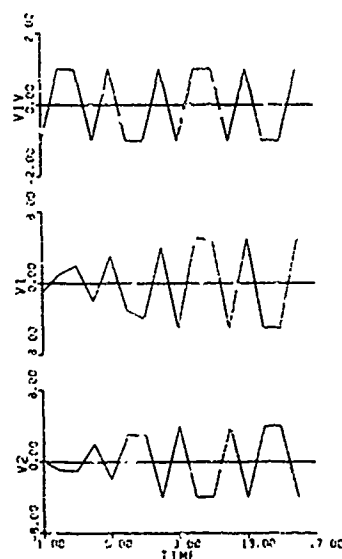


Fig. 9. Filter Tuned to $\text{sal}(3, \theta)$ with $V1V = \text{sal}(3, \theta)$.

The separation property of the digital filter can be seen in Fig. 11 where $V1V = \text{sal}(1, \theta) + \text{sal}(3, \theta)$. The output voltage $V2$ is clearly $\text{sal}(3, \theta)$ which has been separated from its sum with $\text{sal}(1, \theta)$. This result is analogous to the sum of frequencies being separated by a resonance filter in the frequency domain. The difference is that frequency resonance

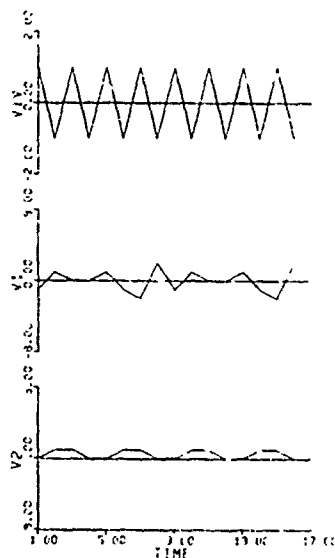


Fig. 10. Filter Tuned to $\text{sal}(3, \theta)$ with $V1V = \text{sal}(4, \theta)$.

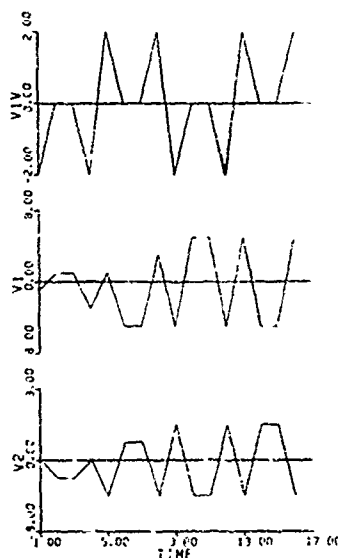


Fig. 11. Filter Tuned to $\text{sal}(3, \theta)$ with $V1V = \text{sal}(1, \theta) + \text{sal}(3, \theta)$.

filters pass a band of frequencies; whereas, the sequency resonance filter passes only a discrete sequency.

Digital resonance filters tuned to other Walsh functions can be developed using these techniques. The simplest version $\text{sal}(4, \theta)$ and the most difficult $\text{sal}(3, \theta)$ were illustrated here. The difficulty of the others lies between these two extremes. The relative ease of

simulating this type of digital filter should make their future very promising.

Derivation of Steady State Difference Equations

The digital simulation of the theoretical model of the LCS serial resonance filter has been presented. The difference equations which were simulated assumed perfect elements with no losses. The simulation of an actual analog model would require that lossy elements be considered in the derivation of the difference equations. To do this, assume that switch S of Fig. 1 when closed has a resistance R and that the initial voltages across the capacitors are $v_1(0) = V_1$ and $v_2(0) = V_2$. Solve for $i(t)$, $v_1(t)$ and $v_2(t)$ for $C_1 = C_2 = C$ with S_0 open if S is closed at $t = 0$. The integro-differential equation for the circuit in Fig. 1 is

$$V_1 - V_2 = L \frac{di(t)}{dt} + \frac{2}{C} \int_0^t i(t) dt + Ri(t). \quad (16)$$

The current and voltages yield the following equations:

$$i(t) = \sqrt{\frac{C}{2L}} (V_1 - V_2) e^{-\alpha t} \sin \omega t \quad (17)$$

$$v_1(t) = V_1 - \frac{1}{2} (V_1 - V_2) \times \left[1 - e^{-\alpha t} \left(\frac{\alpha}{\omega} \sin \omega t + \cos \omega t \right) \right] \quad (18)$$

$$v_2(t) = V_2 + \frac{1}{2} (V_1 - V_2) \times \left[1 - e^{-\alpha t} \left(\frac{\alpha}{\omega} \sin \omega t + \cos \omega t \right) \right] \quad (19)$$

where $C_1 = C_2 = C$, $\omega_0^2 = 2/LC$, $\alpha = R/2L$ and

$$\omega^2 = \omega_0^2 - \alpha^2.$$

The following relations are obtained at the time $t = (\pi/\omega)$:

$$i(\pi/\omega) = 0 \quad (20)$$

$$v_1(\pi/\omega) = \frac{1}{2} (V_1 + V_2) - K$$

$$v_2(\pi/\omega) = \frac{1}{2} (V_1 + V_2) + K$$

where

$$K = \frac{1}{2} (V_1 - V_2) e^{-\alpha\pi/\omega}$$

K must be a positive real number for any physical realizable circuit if $V_1 > V_2$. The voltages across the capacitors at $t = 0$ and $t = \pi/\omega$ no longer interchange as for the ideal circuit. If $R = 500 \Omega$, $f = 400 \text{ Hz}$, $L = 0.21 \text{ Hy}$ and $C = 1.5 \mu\text{F}$ [2], the constant K becomes

$$K = 0.115(V_1 - V_2).$$

Let the initial voltages be $V_1 = 1$ and $V_2 = 0$. The voltages at $t = \pi/\omega$ become

$$v_1(\pi/\omega) = 0.5 - 0.115 = 0.385$$

$$v_2(\pi/\omega) = 0.5 + 0.115 = 0.615.$$

For this value of K, the voltages would slowly build up as time increases. Since K varies exponentially with the amount of resistance, an increase in R would decrease the constant K substantially. For example, suppose $R = 1000 \Omega$; then K would equal 0.052 as compared to 0.115 for $R = 500 \Omega$. This implies that as the series resistance of the circuit approaches infinity the constant K approaches zero. Under this condition, the voltages will not build with time but will remain at a steady state value as long as an input signal at the proper frequency is applied. Upon removal of the input signal the output will damp out very quickly. The point is that an analog LCS serial resonance filter contains a considerable amount of series resistance. The effect of the series resistance is to prevent the continual building up as predicted by the lossless circuit. This would imply that K is small enough to be considered as zero.

The difference equations for this condition can be derived similar to equations (11) which are

$$V_1(J) = V_1(J-1) + 0.5 V_1 V(J-1) \quad (21)$$

$$V_2(J) = V_2(J-1) + 0.5 V_1 V(J-1).$$

The digital simulation under the steady state condition for the filter tuned to $\text{sal}(4, \theta)$ can be obtained by simply substituting equations (21) for the equations (11).

The difference equations for the steady state condition with the filter tuned to $\text{sal}(3, \theta)$ can be derived similar to equations (12) - (15) are

$$\delta_{I+4,J} \begin{bmatrix} V_1(J) = V_1(J-1) + 0.5 V_1 V(J-1) \\ V_2(J) = V_2(J-1) + 0.5 V_1 V(J-1) \end{bmatrix} \quad (22)$$

$$\delta_{I+5,J} \begin{bmatrix} V_1(J) = V_1(J-1) + V_1 V(J-1) \\ V_2(J) = V_2(J-1) \end{bmatrix} \quad (23)$$

$$\delta_{I+6,J} \begin{bmatrix} V_1(J) = V_1(J-1) \\ V_2(J) = V_2(J-1) + V_1 V(J-1) \end{bmatrix} \quad (24)$$

$$\delta_{I+7,J} \begin{bmatrix} V_1(J) = V_1(J-1) + 0.5 V_1 V(J-1) \\ V_2(J) = V_2(J-1) + 0.5 V_1 V(J-1) \end{bmatrix} \quad (25)$$

For steady state digital simulation the equations (22) - (25) would replace the equations (12) - (15).

Conclusion

The LCS resonant filter for Walsh functions was successfully simulated on a general purpose computer. The digital simulation of the resonance filter was based on a set of difference equations which described the lossless theoretical model. Therefore, the simulated output had to be limited to an arbitrary value to mimic the steady state condition of an analog filter. The steady state difference equation for the lossy analog circuit was derived.

The resonance condition is clearly indicated by the filter if and only if the forcing function is at the same frequency as the filter is tuned.

With the advent of integrated circuits analog or continuous Walsh functions may easily be generated and used for carriers in multiplexing or coding systems. Discrete Walsh functions can easily be generated by special purpose or general purpose computers which could also be used for multiplexing or coding.

References

- [1] H. F. Harmuth, "Survey of Analog Sequence Filters Based on Walsh Functions," presented at the 1970 Workshop on Applications of Walsh Functions, Naval Research Laboratory, Washington, D.C., March-April, 1970, AD-707431.
- [2] J. P. Golden and S. N. James, "Implementation of Walsh Function Resonant Filter," IEEE Transactions on Electromagnetic Compatibility, August, 1971, AD-727-000.
- [3] J. P. Golden, Sequence Filters for Walsh Functions, Ph.D. Dissertation, Department of Electrical Engineering, Auburn University, Auburn, Alabama, 1971.
- [4] H. F. Harmuth, Unpublished Notes.

RESONANT SEQUENCY FILTERS IN THE Z-DOMAIN

H. Troy Nagle, Jr.
 Assistant Professor of Electrical Engineering
 Auburn University
 Auburn, Alabama USA

Abstract

In this paper, the characteristics of a resonant sequency filter proposed by Harmuth [1] and constructed by Golden [2] are examined in the z-domain as a digital filter. Special-purpose digital hardware is proposed to realize resonant sequency filters and a BASIC simulation is used to verify the scheme.

Introduction

A digital filter is a discrete-time device which transforms an input data stream into an output data stream by an algebraic process described by constant coefficient difference equations. The transfer function $D(z)$ may be conveniently defined in the z-domain ($z^{-1} = e^{-Ts}$, T is the sampling period, and s the Laplace variable),

$$D(z) = \frac{a_0 + a_1 z^{-1} + \dots + a_n z^{-n}}{1 + b_1 z^{-1} + \dots + b_n z^{-n}}$$

where a_i and b_i are real coefficients.

The digital filter specified by a z-domain transfer function may be physically realized in several ways; general-purpose computers, special-purpose computers [3,4], or hybrid devices [5]. The digital transfer function $D(z)$ may be programmed in several difference equation forms [6,7]. One of the most common forms is the direct form shown below:

$$\begin{aligned} e_0(kT) &= a_0 e_i(kT) + a_1 e_i(kT-T) + \dots \\ &+ a_n e_i(kT-nT) - b_1 e_0(kT-T) - \dots \\ &- b_n e_0(kT-nT) \end{aligned}$$

where k is an integer, $e_i(kT)$ is the filter input, and $e_0(kT)$ is the filter output. The sampling period T is usually omitted to simplify the notation.

Now let us consider the poles and zeroes of $D(z)$. The z-plane is defined as a mapping of the s-plane according to the rule

$$z = e^{Ts}$$

If we define $\omega_s = 2\pi f_s = 2\pi/T$, then the s-plane may be divided into "strips" as shown in Figure 1. One can show that the primary strip in Figure 1 maps into the unit circle of the z-plane of Figure 2. Furthermore, one can show that the region of stability in the s-plane is the left-half plane; in the z-plane, inside the unit circle. An important point to note is that the origin ($s=0$) in the s-plane maps to unity ($z=1$) in the z-plane.

LCS Resonant Filter

The LCS resonant filter proposed by Harmuth [1] and implemented by Golden [2] used a switching scheme to transfer charge between two capacitors. If the voltages on the capacitors are label $x(t)$ and $y(t)$, the difference equations for the filter tuned to $\text{Sa}(3,8)$ are

$$x(n) = i(n-2) + x(n-2)$$

$$x(n+1) = i(n) + x(n)$$

$$x(n+2) = i(n-1) + x(n-1)$$

$$x(n+3) = i(n+1) + x(n+1)$$

$$y(n) = i(n-1) + y(n-2)$$

$$y(n+1) = y(n)$$

$$y(n+2) = i(n) + i(n+1) + y(n-1)$$

$$y(n+3) = i(n+2) + y(n+1)$$

$$n = 1, 5, 9, \dots$$

where $i(n)$ is the input signal. The initial conditions are specified by defining values for:

$$i(-1) = 0$$

$$i(0) = 0$$

$$x(-1) = 0$$

$$x(0) = 0$$

$$y(-1) = 0$$

$$y(0) = 0$$

The difference equations for $x(t)$ may be transformed into the z -domain to yield the following transfer functions:

$$D(z) = \frac{1}{(z-1)(z+1)} \quad t = T, 5T, 9T, \dots$$

$$D(z) = \frac{1}{z-1} \quad t = 2T, 6T, 10T, \dots$$

$$D(z) = \frac{1}{(z-1)(z+0.5+j.866)(z+0.5-j.866)} \quad t = 3T, 7T, 11T, \dots$$

$$D(z) = \frac{1}{(z-1)(z+1)} \quad t = 4T, 8T, 12T, \dots$$

An important point to note is that each $D(z)$ has a pole at $z = 1$. This pole corresponds to integration in the s -domain. Hence, a resonant sequency filter's input signal must have a zero average value.

Special-Purpose Hardware

The analog LCS resonant sequency filter by Harmuth [1] suggests the digital implementation shown in Figure 3. The input signal is sampled by an A/D converter. There are three accumulators (A0, A1, and A2), a scratchpad memory, data routing logic, a sign register, a tuning register, and a match register.

Operation

The operation of the digital resonant filter follows these four phases:

- (1) Setup
- (2) Sampling
- (3) Transient Response
- (4) Sign Detection.

Setup. The setup phase consists of clearing A0, A1, and A2; loading a four-bit tuning word into the tuning register; and loading an eight-bit word into the matching register. The tuning and matching words are shown in Table 1. Also, the sampling rate of the A/D should be set properly.

Sampling. During the sampling phase, the start signal is given and the A/D (analog-to-digital converter) takes eight samples of the input signal $i(t)$. These eight samples are stored in the scratchpad memory and are accumulated in register A0. After all eight values have been added to A0, the register is shifted three places (divide by 8); register A0 may now be used to remove the average value component ($Wal(0,0)$) from the input signal.

Transient Response. At this point the input signal has been properly conditioned for use in registers A1 and A2. The resonant filter has $A1 = A2 = 0$ which causes a transient response as the input signal is applied. The filter is allowed one cycle over the eight input values to allow the transient response to die.

Sign Detection. Once the transient response has died, the filter runs one more cycle over the eight input values while the sign bits of register A1 and A2 are used to fill the sign register. At the completion of this cycle, the match signal goes to "one" if the tuned Walsh function was present in the input signal.

Theory of Operation

The principle upon which the filter is based follows: if the tuning bit is zero:

$$x(n) = x(n-1) + i(n) - A0$$

$$y(n) = y(n-1);$$

if the tuning bit is one:

$$x(n) = y(n-1)$$

$$y(n) = x(n-1) + i(n) - A0.$$

Rather than switch the values of $x(n)$ and $y(n)$ back and forth between A1 and A2, the input samples $i(n)$ are switched to either A1 or A2 as required. The sign of $x(n)$ may be used for detection as will be shown in the following simulation.

BASIC Simulation

A simulation of the digital resonant sequency filter of Figure 3 has been written in BASIC and is shown in Table 2. The simulation uses the tuning words listed in Table 1. Four example runs of the simulation with the filter tuned to $Cal(1,0)$ are shown to verify the theory of operation.

Case 1. Input: $Cal(1,0)$, Tuned: $Cal(1,0)$

n	i(n)	x(n)	
1	-1	-1	Transient Response Phase
2	-1	0	
3	1	1	
4	1	2	
5	1	3	
6	1	-2	
7	-1	-3	
8	-1	-4	
9	-1	-5	Sign Detection Phase
10	-1	4	
11	1	5	
12	1	6	
13	1	7	
14	1	-6	
15	-1	-7	
16	-1	-8	

Case 2. Input: Cal(2,0), Tuned: Cal(1,0)

n	i(n)	x(n)	
8	1	0	
9	1	1	
10	-1	0	Sign
11	-1	-1	Detection
12	1	0	Phase
13	1	1	
14	-1	0	
15	-1	-1	
16	1	0	

Case 3. Input: 2Sal(2,0) + 3Sal(3,0),
Tuned: Cal(1,0)

n	i(n)	x(n)	
8	1	0	
9	-1	-1	
10	5	0	Sign
11	1	1	Detection
12	-5	-4	Phase
13	5	1	
14	-1	4	
15	-5	-1	
16	1	0	

Case 4. Input: $\sum_{i=1}^6 \text{Wal}(i,0)$,
Tuned: Cal(1,0)

n	i(n)	x(n)	
8	0	-4	
9	-2	-6	
10	0	4	Sign
11	-2	2	Detection
12	0	2	Phase
13	6	8	
14	0	-6	
15	-2	-8	
16	0	-6	

Cases 1 and 4 have inputs containing Cal(1,0) and the sign detection phase shows that a match does indeed occur. However, in cases 2 and 3, Cal(1,0) is not present in the input signal so a match does not occur.

Conclusion

This paper has presented a special-purpose digital implementation of a resonant sequence filter. The design has been verified through simulation using the BASIC language. The filter implementation may be used to detect the presence of a Walsh function in sequence of eight contiguous samples of the input signal to the A/D. The design may be easily extended to handle Walsh functions of higher-order.

References

- [1] H. F. Harmuth, Survey of Analog Sequence Filters Band on Walsh Functions, Workshop on Applications of Walsh Functions, NRL, Washington, D. C., March-April, 1970. AD-707431.
- [2] J. P. Golden and S. N. James, Implementation of a Walsh Function Resonant Filter, IEEEETEC, August, 1971.
- [3] H. T. Nagle, Jr., and C. C. Carroll, Organizing a Special-Purpose Computer to Realize Digital Filters for Sampled-Data Systems, IEEEETAE, September, 1968, p. 398-412.
- [4] J. W. Jones, Jr., and C. C. Carroll, A Time-Shared Digital Filter Realization, IEEEETC, November, 1969, p. 1027-1030.
- [5] H. T. Nagle, C. C. Carroll, and J. W. Jones, A Hybrid Realization for Sampled-Data Controllers, IEEEETC, July, 1970, p. 31-37.
- [6] H. T. Nagle, and C. C. Carroll, Memory Sizing for Digital Filters, IFIP Congress 71, Ljubljana, Yugoslavia, August, 1971.
- [7] H. T. Nagle, Jr., Computer Aided Design of Digital Filters, submitted to IEEEETAE, March, 1972, 72 typed pages.

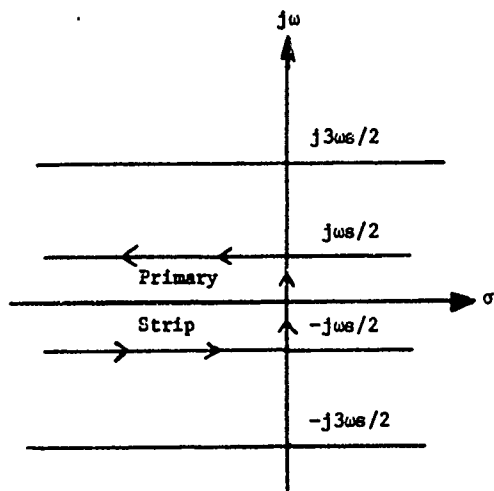


Fig. 1. The s-plane.

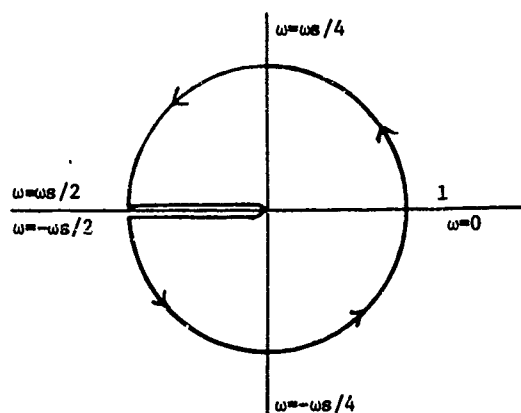


Fig. 2. The z-plane.

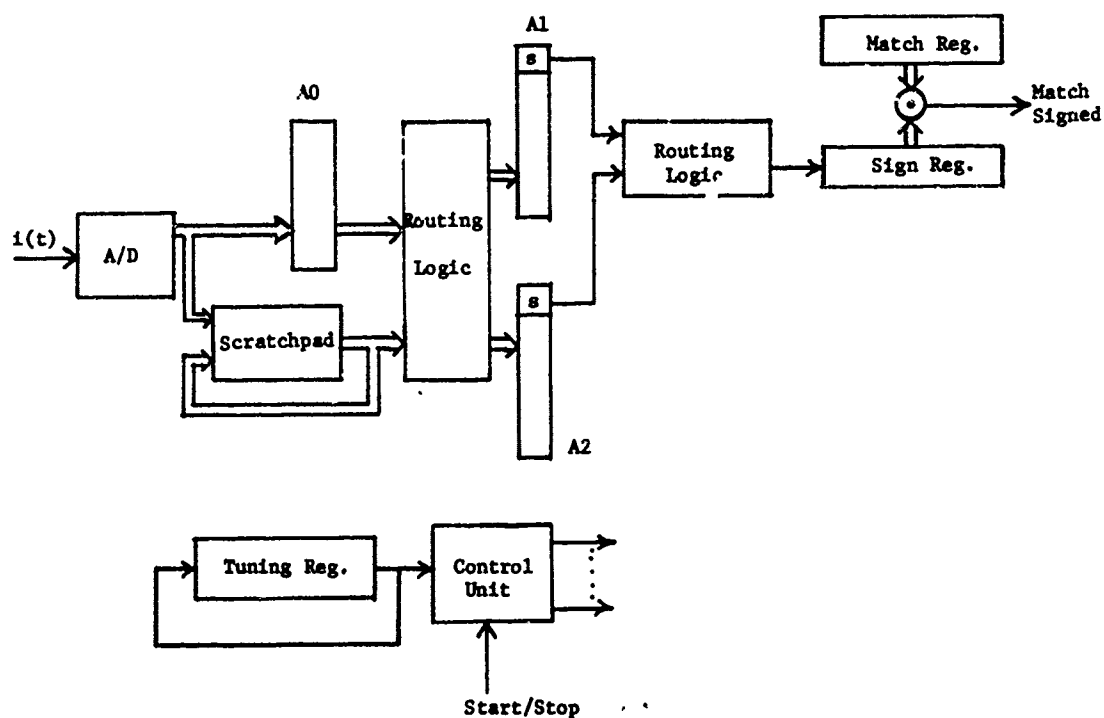


Fig. 3. Digital Resonant Sequence Filter.

Table 1. Tuning and Matching Words

Walsh Function	Tuning Word	Matching Words
	1 2 3 4	1 2 3 4 5 6 7 8
Wal(0,0)	0 0 0 0	0 0 0 0 0 0 0 0
Sal(1,0)	0 0 0 1	1 1 1 1 0 0 0 0
Cal(1,0)	0 1 0 0	1 1 0 0 0 0 1 1
Sal(2,0)	0 1 0 1	0 0 1 1 0 0 1 1
Cal(2,0)	1 0 1 0	0 1 1 0 0 1 1 0
Sal(3,0)	1 0 1 1	1 0 0 1 0 1 1 0
Cal(3,0)	1 1 1 0	1 0 1 0 0 1 0 1
Sal(4,0)	1 1 1 1	0 1 0 1 0 1 0 1

Table 2. BASIC Simulation

```

10 REM SEQUENCY FILTER SIMULATION
20 REM CALL THRU CAL4 FILTERS
30 REM S(1) THRU S(8) ARE TUNING CONSTANTS
40 REM I(1) THRU I(8) ARE INPUT SIGNAL
   VALUES
50 DIM S(8),I(8)
60 PRINT "FILTER CASE ";
70 INPUT C
90 PRINT "INPUT 4 TUNING BITS"
100 FOR N=1 TO 4
110 INPUT S(N)
115 S(N+4)=S(N)
120 NEXT N
130 PRINT
135 T=0
140 PRINT "TYPE 8 INPUT SIGNALS"
150 FOR N=1 TO 8
160 INPUT I(N)
165 T=T+I(N)
170 NEXT N
180 T=T/8
200 Y1=0
205 PRINT
206 PRINT
210 X1=0
220 PRINT "SIMULATION BEGINS"
230 PRINT
280 PRINT "  N  I  X  Y"
290 PRINT
300 FOR M=1 TO 2
310 FOR N=1 TO 8
320 IF S(N)=0 THEN 350
330 X=X1
335 Y=X1+I(N)-T
340 GO TO 360
350 X=X1+I(N)-T
355 Y=Y1
360 PRINT N;I(N);X;Y
365 Y1=Y
370 X1=X
380 NEXT N
410 NEXT M
420 GO TO 135
999 END

```

FREQUENCY DOMAIN DESIGN OF SEQUENCY FILTERS

A. E. Kahveci and E. L. Hall

Department of Electrical Engineering

University of Missouri-Columbia

Columbia, Missouri

Abstract

It is well known that the fast Hadamard transform is computationally advantageous over the fast Fourier transform for filtering operations; however, more design information is available for the frequency domain than for the relatively recent sequence domain. To take advantage of both the simple design techniques of the fast Fourier transform and the decreased computation time of the fast Hadamard transform, the following problem is considered: Given the discrete Fourier transform of a desired filter, derive an equivalent Hadamard sequence filter. This problem is solved for one dimensional filters.

The solution consists of deriving a relation between the frequency and sequence domain filters and then showing that the equivalent sequence filters may actually be implemented with the fast Hadamard transform.

Examples of equivalent filtering of signals are given. The cost of the equivalence filtering is that the normal scalar multiplication may be increased to vector multiplication; however, it is possible to maintain an efficient computation for certain cases.

Introduction

It is well-known that the fast Hadamard transform is computationally advantageous over the fast Fourier transform. However, the Hadamard transform filters are not as easy to design as simple frequency domain fast Fourier transform filters. To take advantage of both the simple design techniques of the fast Fourier transform and the decreased computation time of the Hadamard transform or recursive filter, the following problem is considered: given the discrete Fourier transform of a desired spatial filter, derive an equivalent Hadamard sequence filter.

Pratt [1] has shown that discrete Weiner Filtering may be implemented by any unitary transformation, rather than just the Fourier transform.

The purpose of this paper is to describe methods for designing Hadamard transform filters given the equi-spaced discrete frequency domain specifications. This solution allows one to use the fast Fourier transform for experimentation with the design of a digital filter for signal enhancement, then implement the filter with a faster computational method.

In the next section, equivalent Hadamard transform filtering is described. Then computational problems are considered.

The block diagram of a Fourier filtering system is shown in Fig. 1. For a one-dimensional system, the input column vector, s , is first discrete Fourier (DFT) transformed, producing the DFT of s . Next, a point-by-point multiplication in the frequency domain is performed. Finally, the inverse Fourier transform of the filtered input transform is computed to produce the output column vector, r . Thus, in matrix form:

$$r_1 = F^{-1} G_1 F s \quad (1)$$

where r_1 and s are N length column vectors,

$$F = \frac{1}{\sqrt{N}} [W^{nk}], W = e^{-j2\pi/N}, F^{-1} = F^*, N = 2^m \quad (2)$$

and G_1 is an $N \times N$ filter matrix whose terms are the filter weights. If only the diagonal terms of G_1 are non-zero the filter is called a scalar filter [1]. If at least one off diagonal term is non-zero, the operation is referred to as vector filtering. For most cases a real scalar filter is desired thus G_1 must be conjugate symmetric, i.e., if g_0, g_1, \dots, g_{N-1} are the diagonal elements of G_1 then $g_i = g_{N-1-i}^*$ for $i = 1, \dots, (N/2-1)$.

Now consider the problem of determining a Hadamard transform filter, G_2 , which performs an equivalent operation as the Fourier transform filter, G_1 .

Equivalent Hadamard Filter

A large amount of filter design information for filtering signal via the fast Fourier transform is available; [2,3]; however, Hadamard transform sequence filtering is relatively new, [4,5]. Therefore, implementing a frequency domain filter via a sequence domain computation is economical in both design and computation time.

The block diagram of a filtering system using the Hadamard transform is shown in Fig. 2. Again for the one-dimensional case, s and r are $N \times 1$ column vectors and

$$r_2 = H^{-1} G_2 H s$$

where [1]

$$H = H_N P, H^{-1} = H$$

$$H_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$H_N = \frac{1}{\sqrt{N}} \begin{bmatrix} H_{N/2} & H_{N/2} \\ H_{N/2} & -H_{N/2} \end{bmatrix}$$

$$N = 2^m$$

where P is a row permutation matrix and G_2 is an $N \times N$ filter matrix. For scalar filtering the diagonal terms are the filter weights and the off diagonal terms are zero.

If equivalent outputs are to be produced by both the Fourier and Hadamard systems, then $r_1 = r_2$ or:

$$F^{-1}G_1Fs = H^{-1}G_2Hs$$

$$HF^{-1}G_1Fs = HH^{-1}G_2Hs = G_2Hs$$

Therefore:

$$HF^{-1}G_1F = G_2H$$

or

$$G_2 = HF^{-1}G_1FH^{-1} = HF^{-1}G_1FH$$

Clearly the two filtering results are the same, i.e.

$$r_2 = H^{-1}G_2H = H^{-1}HF^{-1}G_1FH^{-1}H = F^{-1}G_1F = r_1$$

It is interesting to investigate the conditions previously imposed on G_2 , i.e., that it be a real, diagonal matrix for scalar filtering.

An example of the equivalent Hadamard filter computation for $N=4$ is shown in Fig. 3 and illustrates that the G_2 matrix is not necessarily diagonal. In the example G_2 may be transformed into a diagonal matrix simply by row or column transformations.

In general, G_2 is always diagonalizable which may be seen from the following argument.

$$\text{Let } P = F^{-1}G_1F \text{ or } G_1 = FPF^{-1}$$

then since F is a non-singular matrix, P and G_1 are similar [3]. Now since G_1 is a diagonal matrix, the characteristic roots are simply the diagonal elements. Furthermore, since P and G_1 are similar, they have the same characteristic roots.

Now $G_2 = HPH^{-1}$ and since H is non singular, G_2 and P are similar. Thus, G_1 and G_2 have the same characteristic roots. Finally, since G_2 is similar to a diagonal matrix it is diagonalizable.

Note that if G_2 is not a real, diagonal matrix, then the speed of computation may be greatly affected. For example, the point by point multiplication of N points may have to be replaced by an $N \times N$ matrix computation. On the other hand, if G_2 is not a diagonal matrix but

has a large number of zero elements, an efficient computation is possible.

Figure 4 shows an 8 point example. In this example a unity filter was used. As it can be seen, the result of filtering in both domains is equal. The slight difference between the numbers is the roundoff error. In Fig. 5 a band reject filter was used. A signal, $f(nT) = \sin 2\pi 3nt + \sin 2\pi 4nT$, $n=0, \dots, 15$ and $T=1/16$ was filtered of its 3 and 4 H_2 components.

Figure 6 is the printout of the Fourier domain filter. Figure 7 shows the Walsh filter matrix. Fig. 8 is the time domain representation of $f(nT)$.

Figure 9 is the result of filtering in the Fourier domain. As it can be noticed, the amplitude of the filtered signal is very low. It is not completely zero due to Gibbs phenomena. Fig. 10 is the Walsh domain filtered signal. It is exactly equal to the signal in Fig. 9.

Figure 11 shows another example where the input was a series of impulses. Figure 12 shows the low pass filter used. Fig. 13 shows the Fourier filter terms and Fig. 14 shows the corresponding Walsh filter terms. Figure 15 is the Fourier domain filter signal of Fig. 11. Fig. 16 is the Walsh domain filtered signal. As it can be seen again, Fig. 15 and Fig. 16 are exactly the same.

References

- [1] W.K. Pratt, "Generalized Wiener Filtering Computation Techniques," Proceedings of the UMC Two Dimensional Digital Signal Processing Conference, Oct. 1971, pp. 1-11, 1-1-10.
- [2] B. Gold and C.M. Rader, Digital Signal Processing, McGraw-Hill, New York, 1969, pp. 173-181.
- [3] H.C. Andrews, Computer Techniques in Image Processing, Academic, New York, 1970, pp. 105-131.
- [4] J.L. Walsh, "Remarks on the History of Walsh Functions," Proceedings of the Walsh Function Symposium, Washington, D.C., April, 1970.
- [5] W.K. Pratt, "Linear and Non-Linear Filtering in the Walsh Domain", Symposium on the Applications of Walsh Functions, Washington, D.C., April, 1971.

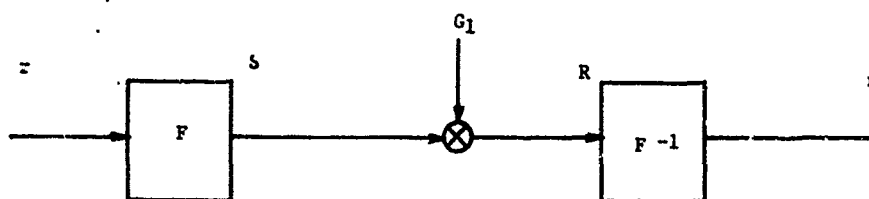


Fig. 1. Filtering via the fast Fourier transform (FFT).

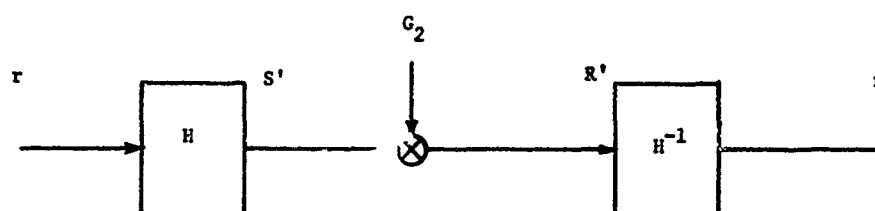


Fig. 2. Filtering via the fast Hadamard transform (FHT).

Fig. 3. Example of equivalent filter computation for $N=4$.

$$G_1 = \begin{bmatrix} g_0 & 0 & 0 & 0 \\ 0 & g_1 & 0 & 0 \\ 0 & 0 & g_2 & 0 \\ 0 & 0 & 0 & g_3 \end{bmatrix}, \quad H = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} = H^{-1}$$

$$F = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}, \quad F^{-1} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & +j \\ 1 & -1 & 1 & -1 \\ 1 & +j & -1 & -j \end{bmatrix}$$

So

$$P = F^{-1} G_1 F = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & +j \\ 1 & -1 & 1 & -1 \\ 1 & +j & -1 & -j \end{bmatrix} \begin{bmatrix} g_0 & 0 & 0 & 0 \\ 0 & g_1 & 0 & 0 \\ 0 & 0 & g_2 & 0 \\ 0 & 0 & 0 & g_3 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}$$

$$P = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & +j \\ 1 & -1 & 1 & -1 \\ 1 & +j & -1 & -j \end{bmatrix} \begin{bmatrix} g_0 & g_0 & g_0 & g_0 \\ g_1 & jg_1 & -jg_1 & -jg_1 \\ g_2 & -g_2 & g_2 & -g_2 \\ g_3 & -jg_3 & -g_3 & jg_3 \end{bmatrix}$$

$$P = \frac{1}{4} \begin{bmatrix} g_0+g_1+g_2+g_3 & g_0+jg_1-g_2-jg_3 & g_0-g_1+g_2-g_3 & g_0-jg_1-g_2+jg_3 \\ \dots & \dots & \dots & \dots \\ g_0-jg_1-g_2+jg_3 & g_0+g_1+g_2+g_3 & g_0+jg_1-g_2-jg_3 & g_0-g_1+g_2-g_3 \\ \dots & \dots & \dots & \dots \\ g_0-g_1+g_2-g_3 & g_0-jg_1-g_2+jg_3 & g_0+g_1+g_2+g_3 & g_0+jg_1-g_2-jg_3 \\ \dots & \dots & \dots & \dots \\ g_0+jg_1-g_2-jg_3 & g_0-g_1+g_2-g_3 & g_0-jg_1-g_2+jg_3 & g_0+g_1+g_2+g_3 \end{bmatrix}$$

Note that P is Hermitian, i.e., $p_{ij} = p_{ji}^*$. Furthermore, if g is symmetric conjugate, i.e., $g_1 = g_3^* = a+jb$ then $(jg_1-jg_3) = -2b$ which is real so that P is real, since both g_0 and g_2 are also real. Under the symmetric conjugate condition on g , P reduces to

$$P = \frac{1}{4} \begin{bmatrix} b_0 & b_3 & b_1 & b_2 \\ b_2 & b_0 & b_3 & b_1 \\ b_1 & b_2 & b_0 & b_3 \\ b_3 & b_1 & b_2 & b_0 \end{bmatrix} \quad \text{where} \quad \begin{aligned} b_0 &= g_0+g_1+g_2+g_3 \\ b_1 &= g_0-g_1+g_2-g_3 \\ b_2 &= g_0-g_2-j(g_1-g_3) \\ b_3 &= g_0-g_2+j(g_1-g_3) \end{aligned}$$

and all b_i are real.

Note also that four g_i values produce only four b_i values. Finally, note that

$$\begin{aligned}
 b_0 + b_1 + b_2 + b_3 &= 4g_0 \\
 b_0 + b_1 - b_2 - b_3 &= 4g_2 \\
 b_0 - b_1 + b_2 - b_3 &= 2g_1(1-j) + 2g_3(1+j) \\
 b_0 - b_1 - b_2 + b_3 &= 2g_1(1+j) + 2g_3(1-j)
 \end{aligned}$$

Now let us compute $G_2 = \text{NPH}$, i.e.

$$\begin{aligned}
 G_2 &= \frac{1}{16} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} b_0 & b_3 & b_1 & b_2 \\ b_2 & b_0 & b_3 & b_1 \\ b_1 & b_2 & b_0 & b_3 \\ b_3 & b_1 & b_2 & b_0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \\
 &= \frac{1}{16} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} b_0 + b_1 + b_2 + b_3 & b_0 - b_3 + b_1 - b_2 & b_0 + b_3 - b_1 - b_2 & b_0 - b_3 - b_1 + b_2 \\ b_0 - b_1 + b_2 - b_3 & b_0 - b_3 - b_1 - b_2 & b_0 + b_3 - b_1 - b_2 & b_0 - b_3 + b_1 - b_2 \\ b_0 + b_1 - b_2 - b_3 & b_0 - b_3 + b_1 - b_2 & b_0 + b_3 - b_1 - b_2 & b_0 - b_3 - b_1 + b_2 \\ b_0 - b_1 - b_2 + b_3 & b_0 - b_3 - b_1 - b_2 & b_0 + b_3 - b_1 - b_2 & b_0 - b_3 + b_1 - b_2 \end{bmatrix} \\
 &= \frac{1}{16} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 2g_1(1+j) + 2g_3(1-j) \\ 4g_0 + 4g_2 + 2g_3(1-j) + 2g_3(1+j) \\ 2g_1(1-j) + 2g_3(1+j) \\ 4g_0 + 4g_2 + 2g_3(1+j) + 2g_3(1-j) \\ -2g_1(1-j) - 2g_3(1+j) \\ 4g_0 + 4g_2 - 2g_3(1+j) - 2g_3(1-j) \\ -2g_1(1-j) + 2g_3(1+j) \\ 4g_0 - 4g_2 - 2g_3(1+j) + 2g_3(1-j) \end{bmatrix}
 \end{aligned}$$

Finally

$$G_2 = \frac{1}{16} \begin{bmatrix} 16g_0 & 0 & 0 & 0 \\ 0 & 16g_2 & 0 & 0 \\ 0 & 0 & 16g_3 & 0 \\ 0 & 0 & 0 & 16g_1 \end{bmatrix}$$

Now if $g_1 = a + jb$, $g_1 + g_3 = 2a$, $g_1 - g_3 = j2b$

So $4j(g_1 - g_3) = -8b$

$$4g_1(2-j) + 4g_3(2+j) = 4(4a+b)$$

Therefore

$$G_2 = \frac{1}{16} \begin{bmatrix} 16g_0 & 0 & -8b & -8b \\ 0 & 16g_2 & -8b & 8b \\ 0 & 0 & 4(4a+b) & 8b \\ 0 & 0 & -8b & 4(4a+b) \end{bmatrix}$$

Thus, G_2 is not diagonal. However, if $b=0$, G_2 is diagonal.

ARBITRARY INPUTS TO THE FILTER:

FINPUT=(1,1,1,1,5,5,5,5)

.....

FOURIER DOMAIN FILTER HEIGHTS

```
1 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0
0 0 1 0 0 0 0 0
0 0 0 1 0 0 0 0
0 0 0 0 1 0 0 0
0 0 0 0 0 1 0 0
0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 1
```

.....

EQUIVALENT WALSH DOMAIN FILTER HEIGHTS (x1000)

```
15.62 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 15.62 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 15.62 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 15.62 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 15.62 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 15.62 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 15.62 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 15.62
```

.....

RESULT OF FILTERING IN FOURIER DOMAIN

FOUT=(1.0,1.0,1.0,1.0,5.0,5.0,5.0,5.0)

.....

RESULT OF FILTERING IN WALSH DOMAIN

FOUT=(1.0,1.0,1.0,1.0,5.0,5.0,5.0,5.0)

Fig. 4. Eight point filtering with unity gain filter.

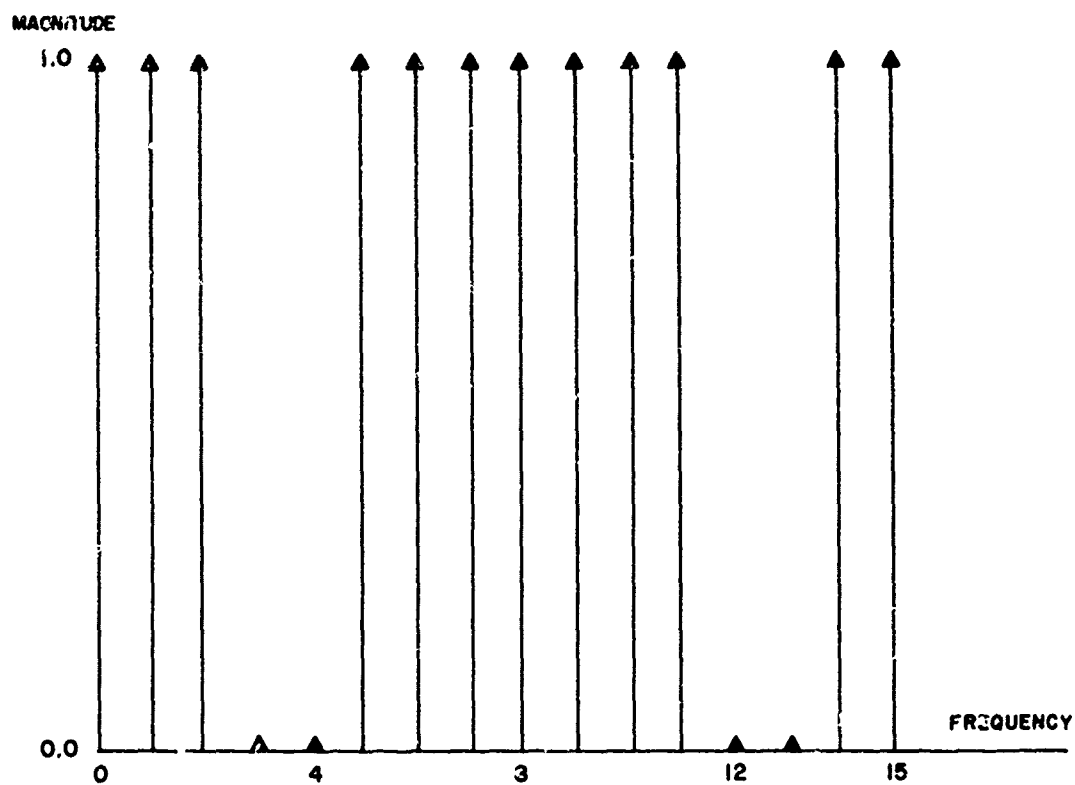


Fig. 5. Sixteen point filter function.

FOURIER DOMAIN FILTER WEIGHTS

```

1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

```

Fig. 6. Fourier domain filter weights.

EQUIVALENT WALSH DOMAIN FILTER WEIGHTS (x1000)

3.91	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	3.62	0.0	0.0	0.0	-.69	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	3.62	0.0	0.0	0.0	0.69	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	2.24	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.69	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	2.24	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-.69	0.0	0.0	0.0
0.0	-.69	0.0	0.0	0.0	2.24	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.69	0.0	0.0	0.0	2.24	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.95	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.95	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.24	0.0	0.0	0.0	0.69	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.24	0.0	0.0	0.0	-.69	0.0
0.0	0.0	0.0	0.69	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.62	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	-.69	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.62	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.69	0.0	0.0	0.0	3.62	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-.69	0.0	0.0	0.0	3.62	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.9]

Fig. 7. Equivalent Walsh domain filter weights (x1000).

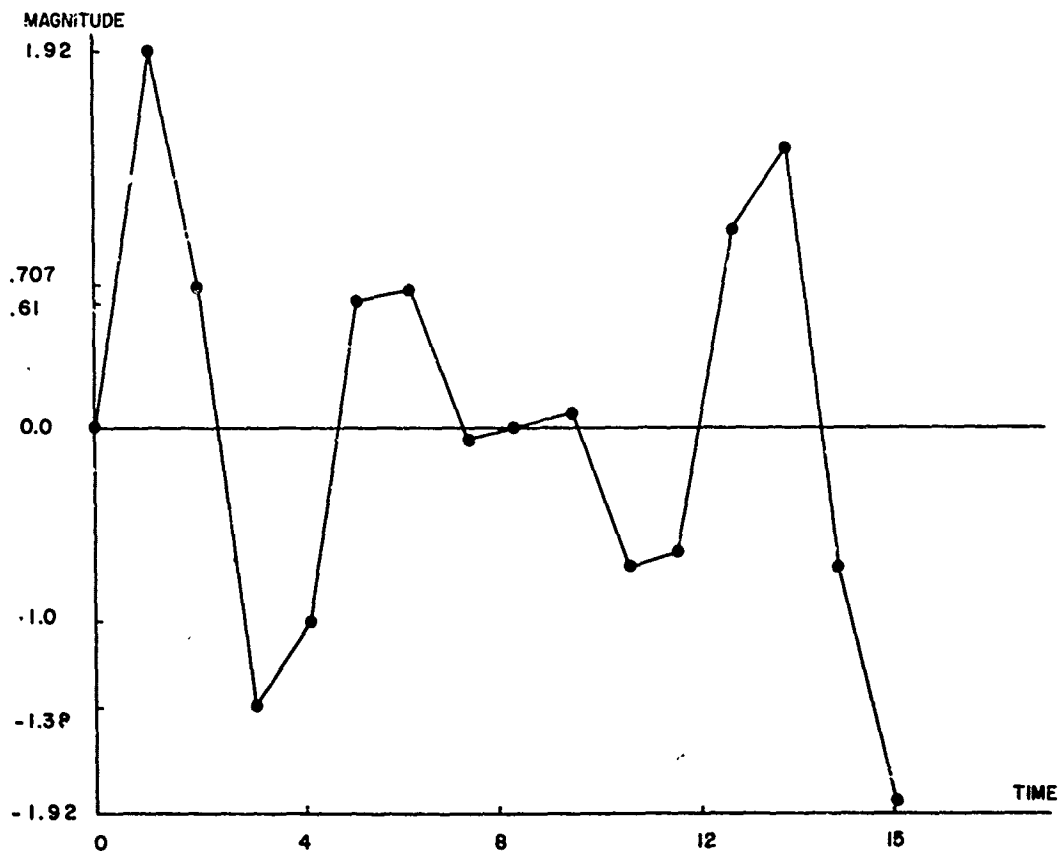


Fig. 8. Input signal to the filter shown in Fig. 5.

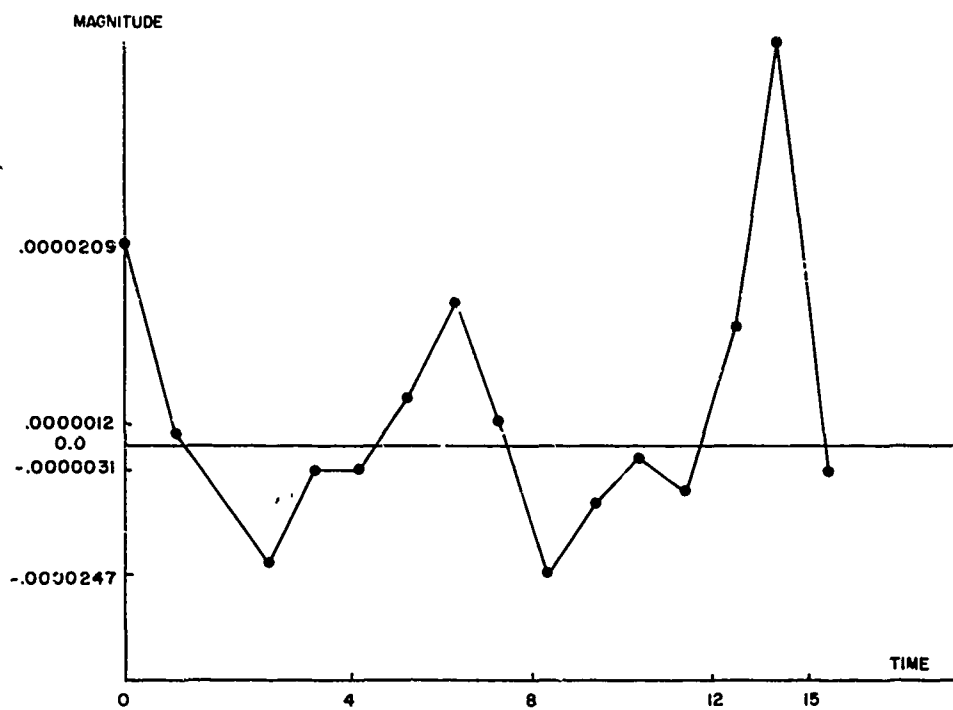


Fig. 9. Fourier domain filtered signal of Fig. 8.

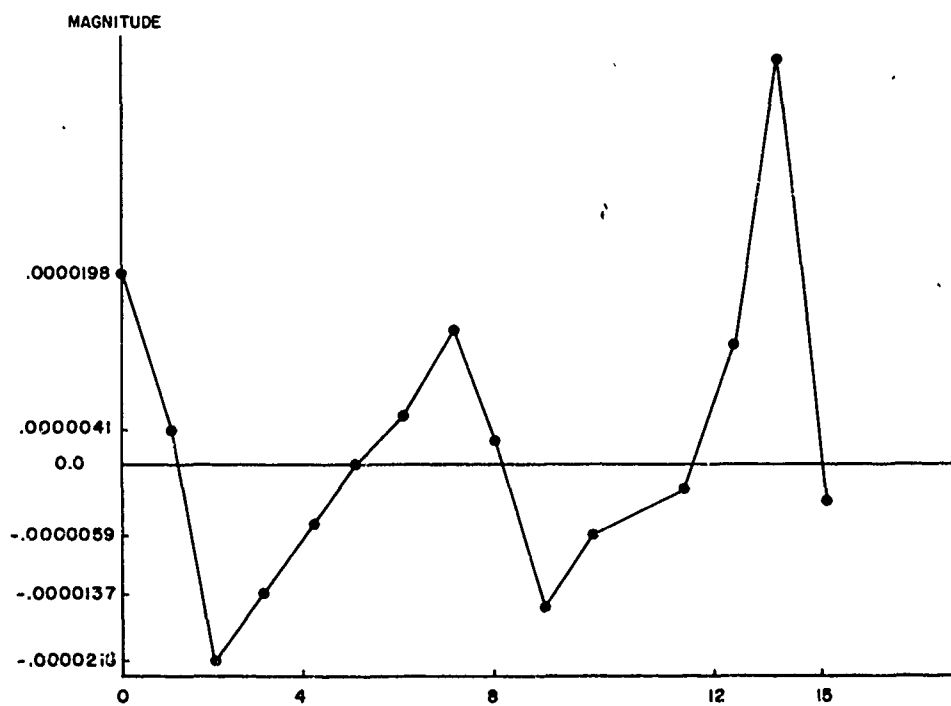


Fig. 10. Walsh domain filtered signal of Fig. 8.

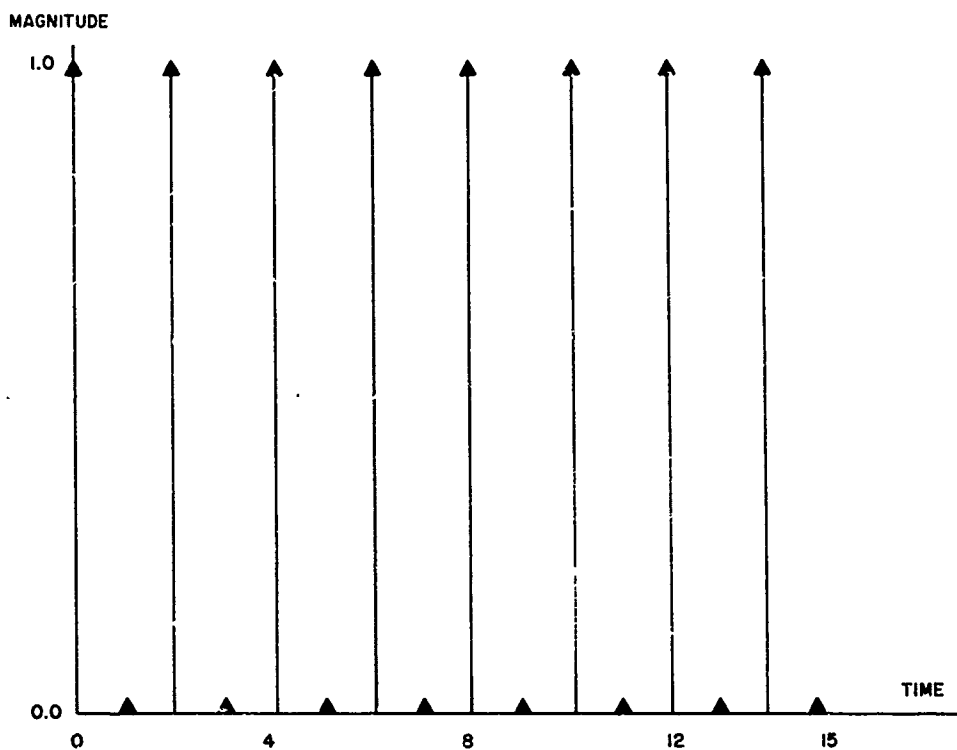


Fig. 11. Input values to the filter shown in Fig. 12.

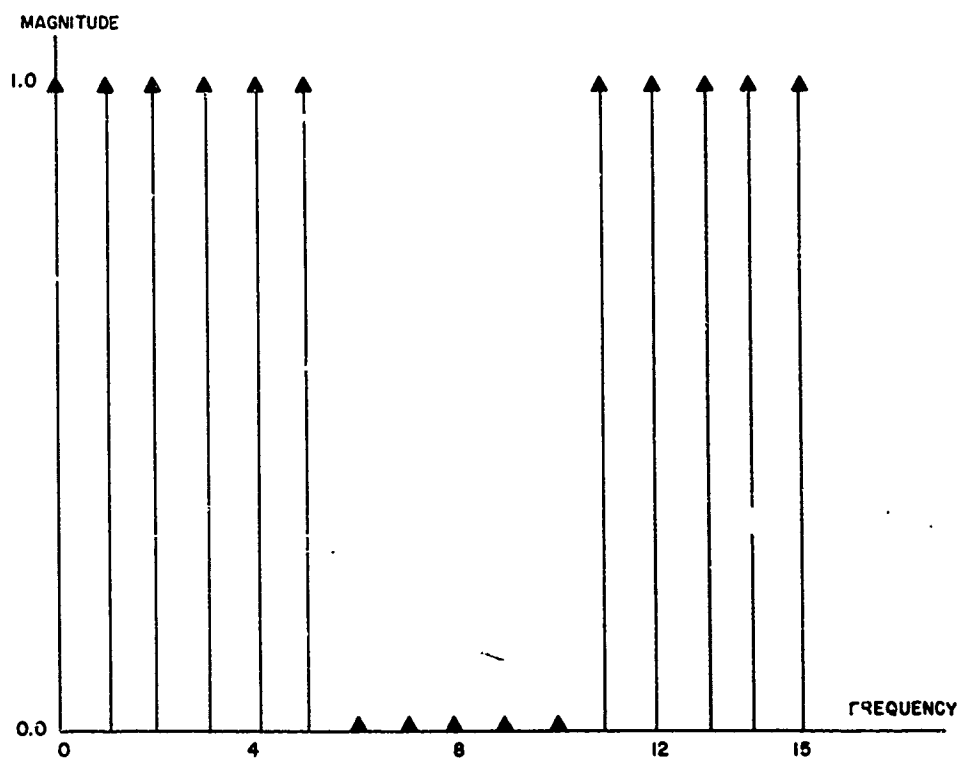


Fig. 12. Sixteen point filter function.

FOURIER DOMAIN FILTER WEIGHTS

```

1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

```

Fig. 13. Fourier domain filter weights.

EQUIVALENT WALSH DOMAIN FILTER WEIGHTS (x1000)

```

3.91 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 3.78 0.0 0.0 0.0 0.05 0.0 0.0 0.0 -0.26 0.0 0.0 0.0 -0.64 0.0
0.0 0.0 3.78 0.0 0.0 0.0 -0.05 0.0 0.0 0.0 -0.26 0.0 0.0 0.0 0.64
0.0 0.0 0.0 3.33 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -1.38 0.0 0.0 0.0
0.0 0.0 0.0 0.0 3.33 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.38 0.0 0.0
0.0 0.05 0.0 0.0 0.0 3.88 0.0 0.0 0.0 0.11 0.0 0.0 0.0 0.26 0.0
0.0 0.0 0.05 0.0 0.0 0.0 3.88 0.0 0.0 0.0 -0.11 0.0 0.0 0.0 0.26
0.0 0.0 0.0 0.0 0.0 0.0 0.0 3.91 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 3.91 0.0 0.0 0.0 0.0 0.0 0.0
0.0 -0.26 0.0 0.0 0.0 0.11 0.0 0.0 0.0 3.36 0.0 0.0 0.0 -1.33 0.0
0.0 0.0 -0.25 0.0 0.0 0.0 -0.11 0.0 0.0 0.0 3.36 0.0 0.0 0.0 1.33
0.0 0.0 0.0 -1.38 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.57 0.0 0.0 0.0
0.0 0.0 0.0 0.0 1.38 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.57 0.0 0.0
0.0 -0.64 0.0 0.0 0.0 0.26 0.0 0.0 0.0 -1.33 0.0 0.0 0.0 0.7 0.0
0.0 0.0 0.64 0.0 0.0 0.0 0.26 0.0 0.0 0.0 1.33 0.0 0.0 0.0 0.7
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

```

Fig. 14. Equivalent Walsh domain filter weights (x1000).

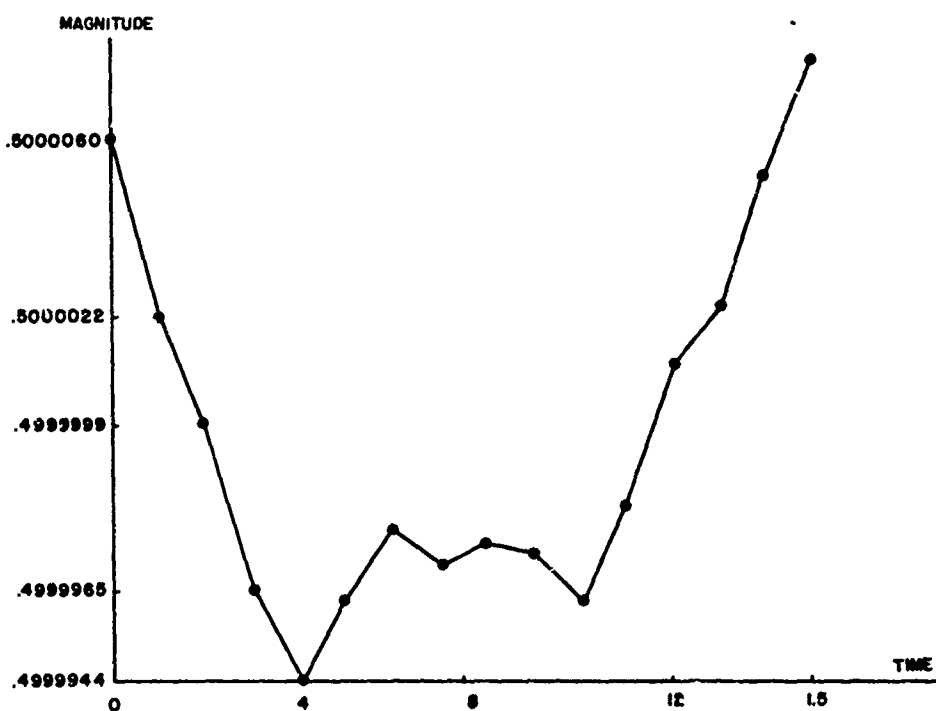


Fig. 15. Fourier domain filtered signal of Fig. 11.

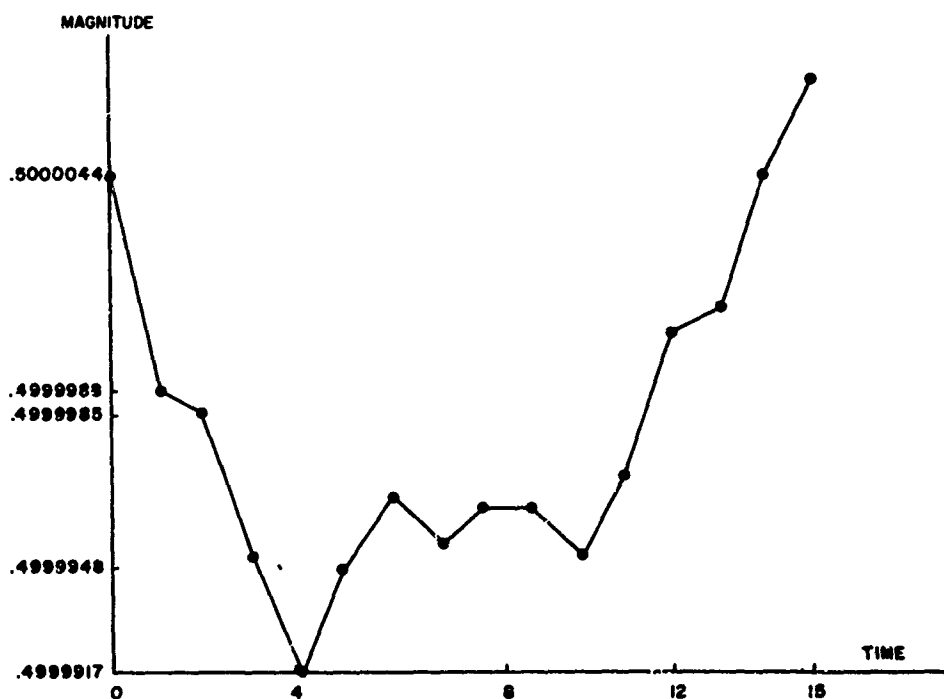


Fig. 16. Walsh domain filtered signal of Fig. 11.

WAVEFORM ANALYSIS OF IMAGE SIGNALS BY ORTHOGONAL TRANSFORMATION

Kosho Shibata
KDD Research Laboratory
Tokyo, Japan

Abstract - By a computer simulation, properties of block waveforms for sampled video signals were examined by the use of an orthogonal transformation, and possibility in compressing frequency bandwidth was examined. As a result, it was found that most of finely divided portions of an image are comparatively simple, number of types is limited, and that they appear on particular waveforms frequently. In the case of a 16th order orthogonal transformation in a 2-dimensional array, when block waveforms of an image are expressed in approximately 1000 types, Sp-p/Nr.m.s is approximately 33 dB, a coding is considered possible at an average of 1.5 bit/sample.

Introduction

Various studies to process images by the use of an orthogonal transformation have been reported.¹⁾²⁾⁴⁾ It seems that the people concerned have used individual statistical properties of transformation outputs to properly quantize orthogonal transformation output.

In this paper, however, examination was made by using relation among individual transformation outputs. Waveforms of fine blocks of a video signal were expressed in a normalized waveform pattern by the use of an orthogonal transformation and the normalized waveform patterns were examined by a computer simulation. As the result, it is understood that types of these patterns are limited and they tend to appear frequently on limited number of particular waveforms. It is expected that these properties of waveform pattern may be used effectively in coding video signals efficiently as well as the various properties for video signal such as correlation between picture elements, prediction error of previous value or linear prediction error level distribution, and frequency spectrum distribution.

In the following, the orthogonal transformation, normalized waveform pattern for block signal waveform, method of computer simulation, the results and discussions therefore are described.

Orthogonal Transform¹⁾³⁾⁴⁾

2-dimensional orthogonal transform of a sample data x_{hi} ($h=1\sim l, i=1\sim m$) is expressed as equation (1). In the case of a 1-dimensional transform, however, it may be expressed as equation (2) because a transformation output is generally used in the form of a transpose.

$$y_{kj} = \sum_{h=1}^l \sum_{i=1}^m b_{kh} x_{hi} a_{ji}^t \quad (1)$$

$$y_{jk} = \sum_{h=1}^l \sum_{i=1}^m a_{ji} x_{hi}^t b_{kh}^t \quad (2)$$

where,

$$\sum_{i=1}^m a_{ji}^2 = 1, \sum_{h=1}^l b_{kh}^2 = 1,$$

$$\sum_{i=1}^m a_{fi} a_{gi} = 0, \sum_{h=1}^l b_{fh} b_{gh} = 0$$

where

x_{hi} : Element of input signal data l, m matrix ($h=1\sim l, i=1\sim m$)

x_{hi}^t : Element of transpose matrix of x_{hi} ; $x_{hi}^t = x_{ih}$

y_{kj} : Element of orthogonal transformation output l, m matrix ($k=1\sim l, j=1\sim m$)

a_{ji} : Element of m th order orthogonal matrix

b_{kh} : Element of l th order orthogonal matrix; $b_{kh}^t = b_{hk}$

Now, when 4th order slant orthogonal matrix MH2S is used for the orthogonal matrix to indicate 2-dimensional orthogonal transform, it is indicated as equation (3).

$$y_{jk} = \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ \frac{3}{\sqrt{5}}(1 & \frac{1}{3} & -\frac{1}{3} & -1) \\ 1 & -1 & -1 & 1 \\ \frac{3}{\sqrt{5}}(\frac{1}{3} & -1 & 1 & -\frac{1}{3}) \end{pmatrix} \begin{pmatrix} x_{11} & x_{22} & x_{31} & x_{41} \\ x_{12} & x_{22} & x_{32} & x_{42} \\ x_{13} & x_{23} & x_{33} & x_{43} \\ x_{14} & x_{24} & x_{34} & x_{44} \end{pmatrix} \begin{pmatrix} 1 & \frac{3}{\sqrt{5}} & 1 & \frac{1}{\sqrt{5}} \\ 1 & \frac{1}{\sqrt{5}} & -1 & \frac{3}{\sqrt{5}} \\ 1 & \frac{1}{\sqrt{5}} & -1 & \frac{3}{\sqrt{5}} \\ 1 & \frac{3}{\sqrt{5}} & 1 & \frac{1}{\sqrt{5}} \end{pmatrix} \quad (3)$$

Fig. 1 shows waveforms lined up in the sequence of this transform after extending the 2-dimensional method to 1-line method.

Among these wave forms, $h_{01}, h_{02}, h_{03}, h_{04}$, and h_{05} respectively indicate DC component, inclination toward vertical direction, inclination toward horizontal direction, horizontal line and vertical line. In the case of a 16th order 2-dimensional Hadamard transform, positive and negative ones in the Fig. 1 may respectively be expressed in 1 and -1. Moreover, equation (4) indicates relation between 4th order Hadamard transformation output y_{0j} and slant orthogonal transformation output y_j .

$$\begin{aligned} y_2 &= \frac{1}{\sqrt{5}}(2y_{02} + y_{04}) \\ y_4 &= \frac{1}{\sqrt{5}}(-y_{02} + 2y_{04}) \end{aligned} \quad (4)$$

Hence, a slant orthogonal transform can be obtained easily from Hadamard transformation output. Equation (5) shows an 8th order 1-line slant orthogonal matrix MH1, and equation (6) indicates relation between transformation output y_j of this MH1 and 8th order Hadamard transformation output y_{0j} .

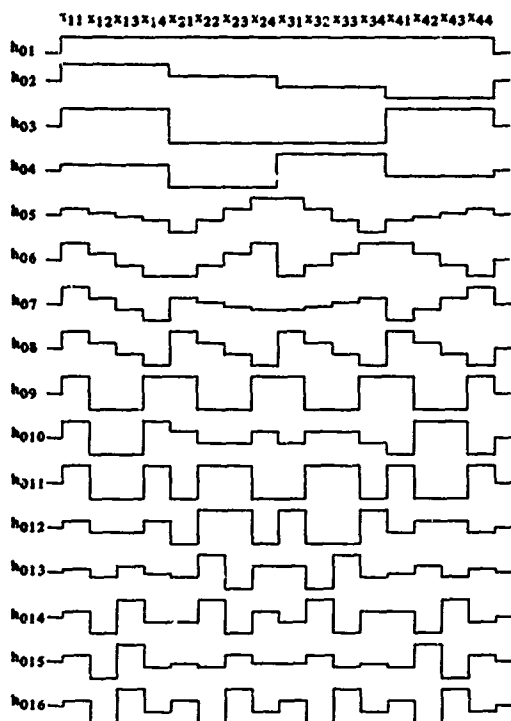


Fig. 1 Waveforms of 16th order 2-dimensional slant transform matrix

$$(h_{ij}) = \frac{1}{\sqrt{8}} \begin{pmatrix} (1 & 1 & 1 & 1 & 1 & 1 & 1 & 1) \\ \frac{7}{\sqrt{3}}(1 & \frac{5}{7} & \frac{3}{7} & \frac{1}{7} & -\frac{1}{7} & -\frac{3}{7} & -\frac{5}{7} & -1) \\ \frac{7}{\sqrt{3}}(1 & \frac{3}{7} & -\frac{1}{7} & -1 & -1 & -\frac{3}{7} & \frac{1}{7} & 1) \\ \frac{17}{\sqrt{5 \cdot 21}}(\frac{7}{17} & -\frac{1}{17} & -\frac{9}{17} & -1 & 1 & \frac{9}{17} & -\frac{1}{17} & \frac{7}{17}) \\ (1 & -1 & -1 & 1 & 1 & -1 & -1 & 1) \\ (1 & -1 & -1 & 1 & -1 & 1 & 1 & -1) \\ \frac{3}{\sqrt{5}}(\frac{1}{3} & -1 & 1 & -\frac{1}{3} & -\frac{1}{3} & 1 & -1 & \frac{1}{3}) \\ \frac{3}{\sqrt{5}}(\frac{1}{3} & -1 & 1 & -\frac{1}{3} & \frac{1}{3} & -1 & 1 & -\frac{1}{3}) \end{pmatrix} \quad (5)$$

$$\begin{aligned} y_2 &= \frac{1}{\sqrt{21}}(4y_{02} + 2y_{04} + y_{08}) \\ y_3 &= \frac{1}{\sqrt{5}}(2y_{03} + y_{07}) \\ y_4 &= \frac{1}{\sqrt{21}}\frac{1}{\sqrt{3}}(-5y_{02} + 8y_{04} + 4y_{08}) \\ y_7 &= \frac{1}{\sqrt{5}}(-y_{03} + 2y_{07}) \\ y_8 &= \frac{1}{\sqrt{5}}(-y_{04} + 2y_{08}) \\ y_1 &= y_{01}, y_5 = y_{05}, y_6 = y_{06} \end{aligned} \quad (6)$$

Waveform Pattern

The orthogonal transform may be considered as a sort of waveform analysis. Input signal waveform is analyzed by orthogonal waveforms, and it is expressed in the form of an output of the waveform components, in other words, as an

output of orthogonal transform. Thus, input signal waveform is regulated depending upon a relation among the transformation outputs. Now, in the case of an n th order orthogonal transform y_1 out of n transformation outputs represents DC component of the input sample waveform. Hence, waveform components excluding the DC component are indicated by y_j other than y_1 . Therefore, when one absolute value of which is maximum is expressed y_m , the y_j is normalized with the y_m to obtain y_j/y_m and y_j/y_m in quantity of $(n-1)$ are treated as one, relation among transformation outputs can be regulated. This is considered as a normalized waveform. Quantized waveform pattern can be obtained by quantizing this normalized waveform at proper levels such as 0, $\pm 1/3$, $\pm 2/3$ and ± 1 . This means that input block waveform is expressed in the form of a normalized pattern. Hereinafter, this pattern is called waveform pattern. This waveform pattern corresponds to input signal waveform in one to one. Thus, input signal can be recovered from this waveform pattern. To be more specifically, input signal can be obtained by multiplying y'_m (quantized y_m) to normalized pattern to obtain each y'_j (quantized y_j) and by conducting inverse orthogonal transform on it after adding y'_1 (quantized y_1).

Computer Simulation

Method of simulation

In order to examine properties of block waveforms of a video signal, various examinations were made by obtaining waveform patterns on waveforms of three types of input signal data block 1×8 , 2×4 and 4×4 arranged on a straight line and plane. For the orthogonal transformation matrices, 2nd, 4th and 8th order Hadamard matrices H^2 , H^4 and H^8 and 4th order slant matrix MH2S and 8th order slant matrix MH1 were used.

For the input video signals, three types of signal data A, B and C obtained from three types of SMPTE slides; (A) close-up, (B) medium and (C) fine picture were used. The dynamic range is from 0 to 63, quantizing in 64 levels. For the data A, numbers of blocks for 8th and 16th orders are respectively 2464 and 1232, for the data B, 3315 and 1536, and for data C, 3087 and 1440.

First of all, orthogonal transform was conducted on each block of input data by the use of H, MH1 for the 8th order straight arrangement to obtain maximum value y_m of transformation output other than DC component and ratio of (y_j/y_m) ; waveform pattern (y_j/y_m) ($j=2 \sim 8$) was obtained by quantizing it; and thus, types and appearance frequency were obtained. Amplitude distribution of y_m , ratio y_m/y_1 and frequency on each y_j of y_m were also obtained. A degree of waveform distortion due to quantizing was examined by obtaining quantizing noise. For the quantizing noise, y'_m obtained by quantizing y_m was multiplied to the waveform pattern to obtain y'_j ($j=2 \sim n$); y'_1 obtained by quantizing y_1 was added to it to conduct inverse orthogonal transform, recovery signal x'_i was thus obtained; mean square error of quantization was obtained with difference between the recovery signal x'_i and input signal x_i ; and thus, Sp-p/Nr.m.s was obtained. In addition, linear prediction error $\hat{x}_i = \frac{x_{i-1} + x_{i+1}}{2} - x_i$ of input data x_i was obtained, and Sp-p/Nr.m.s of the quantizing error against the value of \hat{x}_i was obtained. For the case of 8th order, y_1 and y_m were respectively quantized by 7 and 5 bits,

and y_j/y_m was quantized with 0, $\pm 1/3$, $\pm 2/3$ and ± 1 after making y_j zero under the following provisions:

$$\begin{aligned} \text{For } y_2 \sim y_4 : y_j < 1 \\ \text{For } y_5 \sim y_8 : y_j < 2 \end{aligned}$$

In the case of 16th order, y_1 and y_m were quantized with eight and six bits, and as for $y_2 \sim y_8$, y_j was made to zero when $y_j < 1$; and thus, y_j/y_m was quantized in the same manner as the case of 8th order. As for 2-line array data of 8th order, when transformation matrix is a Hadamard matrix, (H^4 , H^2) was examined, and MH2 ($MH2S$, H^2) was examined in the case of 2-line inclination, both in the approximately same manner as the case of 1-line array. In addition, for 16th order 4-line array data, (H^4 , H^4) was examined when transformation matrix is Hadamard, and ($MH2S$, $MH2S$) was examined in the case of plane inclination for the same matters.

Result of simulation and discussion

(1) Result of simulation on H, MH1

For video data A, B, and C under linear array, 8th order slant orthogonal transform MH1 was examined. Fig. 2 shows relative level distribution of ratio y_m/y_1 between block DC component y_1 and maximum change component y_m , and Fig. 3 shows percentage of y_m appearance frequency on each y_j . Distributions of y_m/y_1 for three data are mutually similar. In the case of A (close-up), however, zero appears remarkably frequently, the overall value is smaller than that of B and C data, and it reveals a tendency

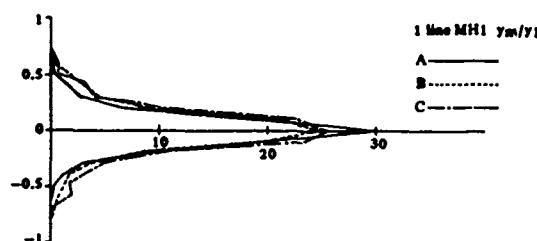


Fig. 2 Distribution of ratio y_m/y_1
(Horizontal axis: Probability density, $\times 10^{-2}$)

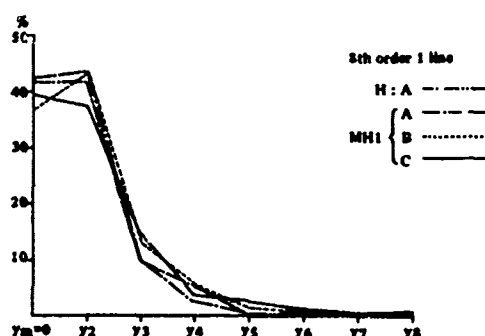


Fig. 3 Percentage of appearance frequency of y_m on each y_j

that values are greater in the order of B and C data. As for percentage of y_m on each transformation output, y_2 appears most frequently (approximately 40%) for all A, B and C data, y_3 is approximately 10%, and for other y_j , it rapidly reduces as j increases. This means that, as far as picture is concerned, component which frequently fluctuates is very minor. Fig. 4 and 5 respectively show level distribution of overall y_m , and level distribution of y_m where y_m is separated to each y_j . In the Fig. 4, levels are higher in the order of A, B and C. In the Fig. 5, amplitude level of y_m reduces as j of y_j increases; frequency at low level is remarkable for A; and B and C are mutually similar. As it is expressed in the Fig. 3, y_m does not appear so frequently where j is great; amplitude is small; and also for the matter of frequency, component is extremely high. Thus, it may be disregarded in processing video signal.

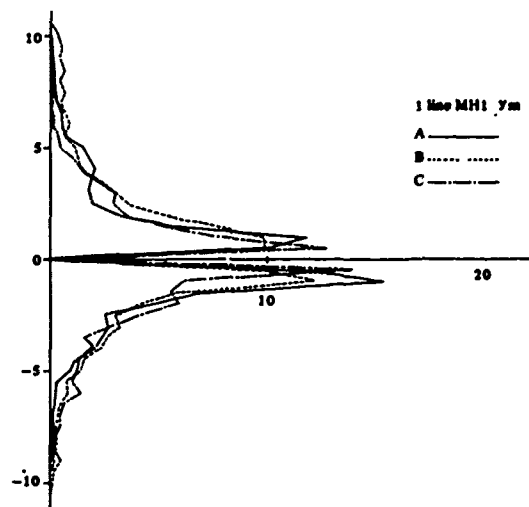


Fig. 4 Level distribution of y_m
(Horizontal axis: Probability density, $\times 10^{-2}$)

Fig. 6 shows Sp-p/Nr.m.s against linear prediction error on H and MH1, together with the ordinary Sp-p/Nr.m.s. When linear prediction error is zero, MH1 is approximately 1.5 dB better than H, it is approximately 4.5 dB better than ordinary ones, and it indicates that quantizing noise is small where fluctuation is minor. Moreover, this indicates that inclination component can be easily expressed by using slant orthogonal matrix. Signal-to-noise ratio is considerably worsened at a part where linear prediction error is large, and it is considered that this may not be regarded because of the visual characteristics. However, care should be exercised on error in edge position.

(2) Property of waveform pattern

For H and MH1 of 8th order 1-line array, H and MH2 of 8th order 2-line array and H and MH2S of 16th order 4-line array, Table 1 shows number of appeared waveform patterns, Sp-p/Nr.m.s and number of input blocks for A, B and C, and the grand total. The value of Sp-p/Nr.m.s is applicable for the case in which signal is reproduced to the original signal by using appeared waveform pattern.

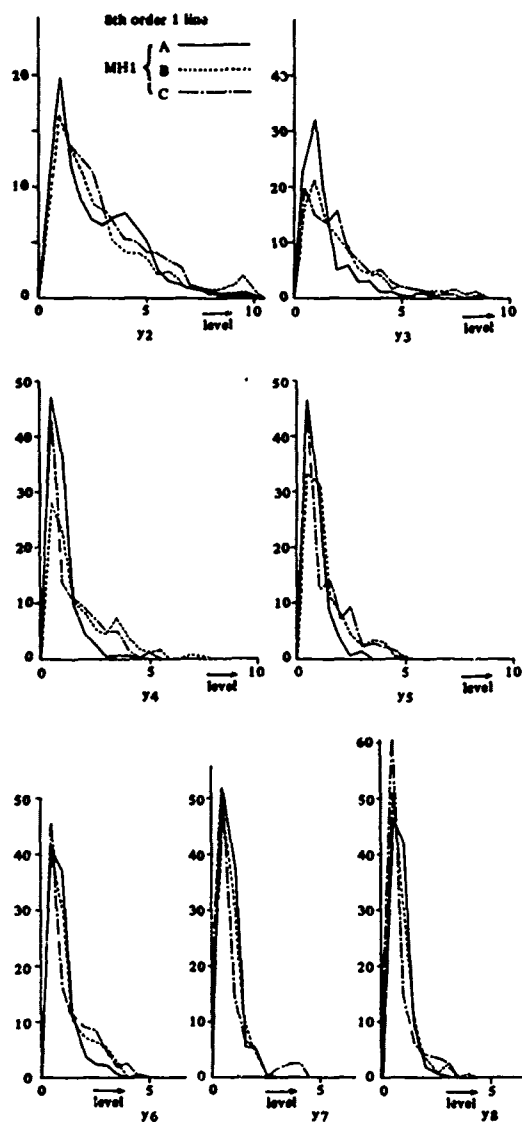


Fig. 5 Level distribution of y_m on each y_j
(Vertical axis: Probability density, $\times 10^{-2}$)

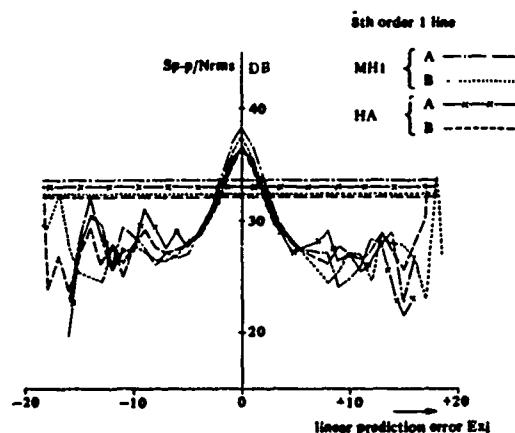


Fig. 6 Sp-p/Nr.m.s for linear prediction error of x_i

When the H of the 8th order 1-line array is compared to the MH1 of the same based on the Table 1, pattern appears less frequently for the MH1 throughout the data A, B and C and the MH1 is slightly better than the H in Sp-p/Nr.m.s for the data A and B, but for the data C (fine picture), the H is slightly better than the MH1. As long as number of patterns is concerned, the MH2 is slightly less than the H also for the case of 2-line array, and signal-to-noise ratio is also improved slightly. As for the comparison between 1-line array and 2-line array, number of waveform patterns considerably reduces and signal-to-noise ratio is slightly improved.

Table 1 Number of types of waveform pattern and S/N

		A		B		C		Grand total
		No of appeared patterns	Sp-p/ Nr.m.s dB	No of appeared patterns	Sp-p/ Nr.m.s dB	No of appeared patterns	Sp-p/ Nr.m.s dB	
No of input data blocks		2464		3315		3087		8866
1-line 8th order	H	209	33.0	447	32.1	365	32.8	1021
	MH1	182	33.6	373	32.3	354	32.4	909
No of input data blocks		2464		3315		3087		8866
2-line 8th order	H	177	33.8	321	33.4	250	33.8	748
	MH2	160	34.3	293	33.6	229	34.2	682
No of input data blocks		1232		1536		1440		4208
4-line 16th order	H	239	33.3	482	32.9	458	33.6	1179
	MH2S	234	33.6	488	33.1	485	33.8	1207

For the H and MH1 of 8th order 1-line, H and MH2 of 8th order 2-line array, and H and MH2S of 16th order 2-dimensional array on the data A, accumulation percentages were obtained in order of higher waveform pattern appearing frequency. Fig. 7 shows the accumulation percentages for the number of waveform patterns obtained as above.

For all cases, it is common that the pattern appears most frequently when input (waveform (h_j/h_m) of which is zero) is flat. In the 1-line array, flat ones are only 40%, but it increases to approximately 60% under 2-dimensional array. Moreover, for both the 2-line and 4-line arrays, Hadamard matrix is almost similar to slant matrix for the matter of accumulated percentage of pattern is concerned. However, in the 1-line array, the slant matrix is highly

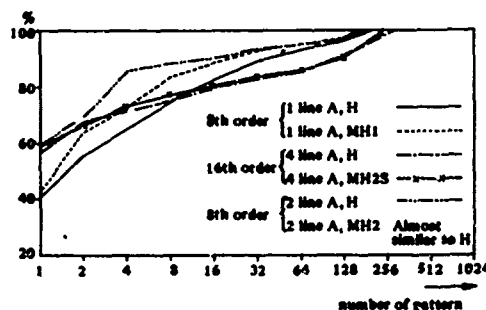


Fig. 7 Accumulated appearance frequency of pattern for number of patterns

concentrated, in comparison with Hadamard matrix, at a portion where number of patterns is small. For the data A, 16 patterns having higher appearance frequency were selected from those on the 8th order 1-line MH1. Fig. 8 shows the patterns and waveforms selected in order of higher appearance frequency. According to the patterns shown on the Fig. 8, y_s/y_m through y_8/y_m are all zero, and throughout the patterns, number of zeros is extremely many. Majority of waveform patterns carry zeros in their last two or three positions. When observing each pattern, appearance frequency of pattern (0,0,0,0,0,0) occupies 42.3%, slant portion (1,0,0,0,0,0) is 21.4%, and generally speaking, it may be said that 4 through 16 express rising and lowering edges. Moreover, (0,0,0,0,0,0),

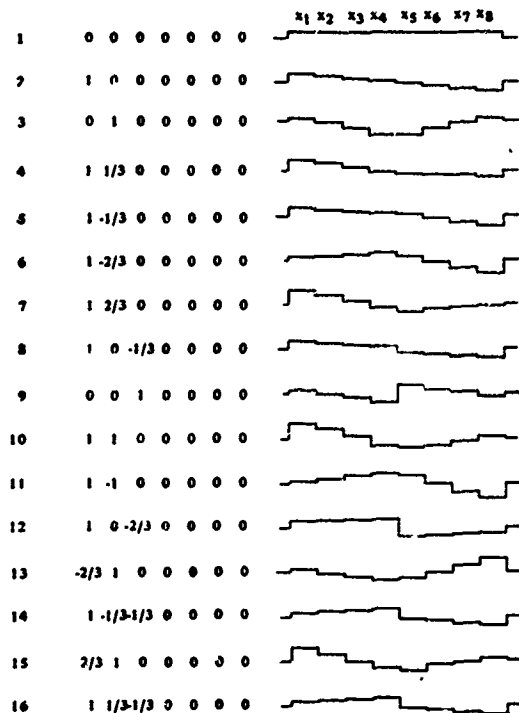


Fig. 8 Waveform patterns and their waveforms

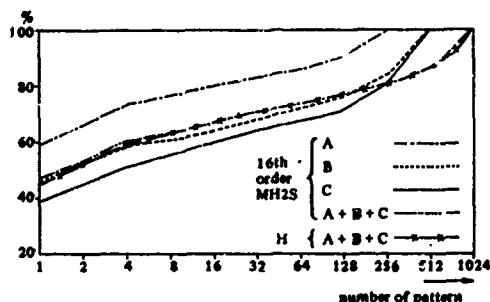


Fig. 9 Accumulated appearance frequency of pattern for number of patterns

Table 2

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	2/3	0	0	0	1	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
6	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
7	1	0	2/3	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	2/3	0	0	0	1	2/3	0	0	0	0	0	0	0
9	0	0	1/3	0	0	0	1	0	0	0	0	0	0	0	0
10	0	0	2/3	0	0	0	1	-2/3	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
12	1	0	0	0	0	0	-2/3	0	0	0	0	0	0	0	0
13	-2/3	0	0	0	0	0	1	0	0	0	0	0	0	0	0
14	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
15	1	0	1/3	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	2/3	0	0	0	1	0	0	0	0	0	0	0	0

(1,0,0,0,0,0), (0,1,0,0,0,0) and (0,0,1,0,0,0) which correspond to waveforms of row vectors of transform matrix appear frequently in order of 1, 2, 3 and 9, and these four take large percentages.

Fig. 9 shows accumulated appearance frequencies for number of patterns on the data A, B, and C for the 16th order MH2S in the same manner as the Fig. 7, and frequencies of H and MH2S on total of the data A, B and C. According to the Fig. 9, appearance frequency of flat ones fluctuate from 40 to 60% depending upon types of input data, and others indicate approximately similar tendency wherein they do not indicate remarkable fluctuation. The totals are respectively 988 and 917, which indicate that the MH2S is slightly less than H, and when examining the grand totals 1179 and 1207 each shown on the Table 1, it indicates reverse symptom. It appears evenly in portions where number of patterns is minor, these patterns are mutually common, and it is assumed that when number of patterns increases, number of types of pattern increases because they appear independently. Table 2 shows 16 patterns which appear in order of higher appearance frequency for total of data A, B and C. According to the Table 2, those of higher appearance frequency are extremely simple, in which all of them have three or less quantizing levels other than zero; h_{00} and h_{01} (Fig. 1) which express horizontal and vertical inclination components are in (2) and (3), and h_{02} and h_{03} which indicate vertical and horizontal lines are in (5) and (14). Most of others include horizontal or vertical inclination components. According to the Table 1, H does not differ from the MH2S as long as noise-to-signal ratio is concerned, however, it may be expected that the 2-line linear prediction error may be improved slightly.

(3) Application to video signal coding

It was found that fine block waveform of video signal can be expressed in a limited number within a range in which error is not remarkable. Now, for example in the case of 16th order orthogonal transform, number of types of waveform pattern is limited to within 1000, if they are coded at 10 bits, and y_1 and y_m are quantized

respectively at 8 and 6 bits, total number of bits is 24, and thus, 1.5 bit/picture element coding may be possible.

Quantizings to obtain waveform patterns were made evenly for all, and it is expected that signal-to-noise ratio will be improved by examining each one of y_j/y_m distribution finely and by suiting them to the individual quantizings. Moreover, in the quantizing, various waveforms possibly appear depending upon input data, and therefore, clustering should be made adequately to prevent deterioration of output image.

Conclusion

Block waveforms of three types of video signals were examined under the normalized waveform method with orthogonal transform. As the result, it was insured that number of types of appearing waveform pattern is considerably limited and that flat and slant portions occupy a large percentage. In this study, mainly number of waveform patterns was examined. It is considered, however, that waveform patterns which express features of block waveforms of video signals can be selected more precisely by making quantization in response to distribution of ratio of each transformation output against the maximum value and thus, video signal can be coded efficiently. In this study, reproduced picture was evaluated in Sp-p/Nr.m.s. However, it should be experimentally insured to prevent deterioration of quality of picture.

Finally, thanks are due to Prof. Enomoto of Tokyo Institute of Technology who provided valuable advice, and personnels of KDD Research Laboratory who supported in proceeding this research.

References

1. W.K. Pratt, J. Kane, and H.C. Andrews, "Hadamard Transform Image Coding," Proc. IEEE 57, No.1, 58-68, Jan. 1969.
2. A. Habibi, and P.A. Wintz, "Image Coding by Linear Transformation and Block Quantization," IEEE Trans. Communication Technology, COM-19, 50-62 (1971).
3. H.F. Harmuth, Transmission of Information by Orthogonal Functions, Springer Verlag, New York 1969.
4. H. Enomoto, and K. Shibata, "Orthogonal Transform Coding System for Television Signals," IEEE Trans. Electromagnetic Compatibility, Special Issue on Walsh Functions Symposium 1971, EMC-13, 11-17, 1971.
5. H. Enomoto, and K. Shibata, "Features of a Hadamard Transformed Television Signals," 1965 National Conference of the I.E.C.E. of Japan 881.
6. H. Enomoto and K. Shibata, "Television Signal Coding Method by Orthogonal Transformations," 1966 Joint Convention of the I.E.C.E. of Japan, 1430.

WALSH FUNCTIONS FOR IMAGE ENHANCEMENT

by

R. L. Richardson, F. R. Reich, and B. P. Hildebrand
Battelle Pacific Northwest Laboratories
Richland, Washington

Introduction

There are a number of important photographic measurement techniques that must rely on shadow imaging. These techniques involve non-optical radiation such as X-rays, γ -rays and neutrons; radiation for which lenses are not available. Images of this type are never sharply focused and hence some form of edge sharpening is desirable in order to be able to make reliable measurements from the photographs. Such enhancement has been performed with coherent optical systems and with computers [1,2]. The optical systems are, in a sense, more natural since they are in themselves image forming. However, they are restricted to linear operations, whereas computers are not.

Computers, despite the fact that they are not restricted to linear processing, have for the most part been used in a linear mode simply because imaging systems operate linearly. Until the rediscovery of the Fast Fourier Transform algorithm, picture processing was impractical because of the long processing times required [3]. Only recently has attention been turned to transforms other than Fourier for picture processing [4]. Even so, no real application of these transforms to picture processing has been made; it has only been demonstrated that they can be used to transform an image and recover it by the inverse transform.

In the real world of photographic recording we are still restricted to the sine-cosine functions. That is, the physics of wave propagation restricts us to Fourier analysis rather than, say, Walsh or Haar function analysis. In

the computer world, however, we would like to work with binary functions rather than continuous function. If we can marry the two worlds, we may be able to perform computer processing of images with simple matrix multiplications in place of the more complicated FFT algorithm at a savings in time and cost.

In this paper we attempt a first step in this direction while addressing ourselves to a real problem. In the nuclear industry a major problem is the damage to fuel rods caused by the intense radioactivity in the reactor core. The way in which this damage is studied is by X-ray or self radiation imaging. Since lenses cannot be used, and point sources of X-ray are not available, sharp shadows or edges are not well defined in the image. What we wish to do then is to sharpen or enhance such an image.

Optical Filtering of Extended Source Shadow Images

It is well known that coherent optical systems can be used to perform Fourier transform operations on two-dimensional input data [1]. These systems are based upon the observation that a picture placed in the front focal plane of a lens will have its two-dimensional Fourier transform displayed in the back focal plane. A mask, varying in transmission and/or thickness, placed in the back focal plane can therefore be used to modify the data in some prescribed manner. That is, the Fourier transform of the data can be multiplied by a filter function as is customary in electronic systems. A second lens can then be used to take an inverse transform to yield a modified image. Such a system is shown in Figure 1.

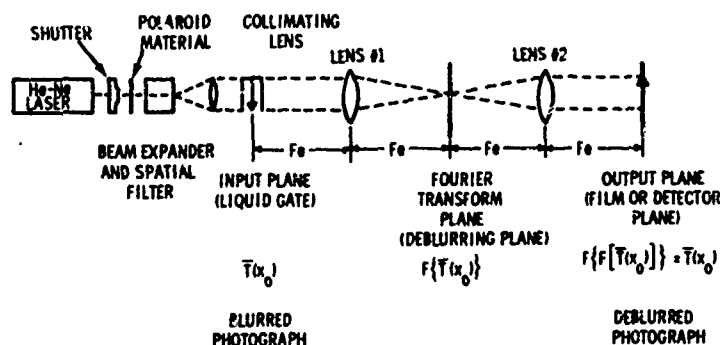


FIGURE 1. COHERENT OPTICAL PROCESSING SYSTEM FOR PICTURE PROCESSING

Neutron and X-ray imaging systems form their images by shadow casting an energy absorbing object with incoherent illumination. The size of the image will depend on the source-to-object and object-to-recording plane distance ratio. If the illuminating source dimensions approach or appear to project from a true geometrical point, a perfect shadow of the absorbing object will be obtained (Figure 2-a) with a large intensity gradient along the edges. If, however, the source appears to emanate from a large area composed of a multiple of true point sources, the image edges will contain a penumbra or unsharpness with small intensity gradients along the edges (Figure 2-b). A one-dimensional analysis of the shadow image intensity shows that the image obtained from an extended source is the convolution of the extended source $\omega(\beta)$ with the image that would be obtained from a true point source $I(x)$ [1].

$$I_e(x) = C \int_{-A}^{A'} \omega(\beta) I(x-\beta) d\beta \quad (1)$$

where A, A' are the source dimensions projected

to the image plane, $\omega(\beta)$ and $I(x)$ are the source intensity distribution and object intensity transmission. The Fourier transform of Equation (1) indicates that the spatial frequency content of $I_e(x)$ will be less than or equal to the true point source shadow image $I(x)$ for all spatial frequency components.

$$F \{I_e(x)\} = CF \{\omega(x)\} F \{I(x)\} \leq F \{I(x)\} \quad (2)$$

To demonstrate this, consider an incoherent line source of dimension L and a source-to-object, object-to-film plane ratio d . From Equation (1), the resultant image spatial frequency content from the Fourier transform operator will be

$$F \{I_e(x)\} = K \frac{\sin \alpha}{\alpha} F \{I(x)\} \quad (3)$$

where K is a constant of the imaging geometry and α is $2\pi d L f$ where f is spatial frequency. The function (Figure 3-a) shows zero points at some spatial frequencies and regions of contrast reversal (regions of $\sin(x)/x$ that are below the zero axis).

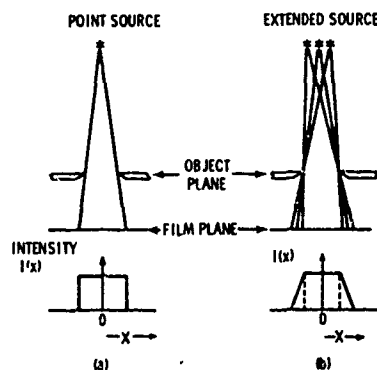


FIGURE 2. PENUMBRA OR UNSHARPNESS OF EXTENDED SOURCE SHADOW IMAGES
(a) SHOWS THE SHAPE OF THE DENSITY FUNCTION PRODUCED ON FILM WHEN A POINT SOURCE IS USED AND (b) WHEN AN EXTENDED SOURCE IS USED

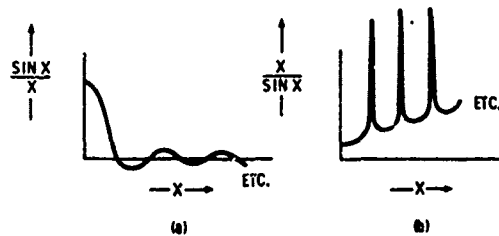


FIGURE 3. (a) INDICATES THE SPATIAL FREQUENCY DISTORTION CAUSED BY AN EXTENDED SOURCE FOR SHADOW IMAGING
(b) SHOWS THE AMPLITUDE OF THE FILTER FUNCTION REQUIRED TO COUNTERACT THIS DISTORTION

In a coherent imaging system (Figure 1), we can make corrections for the extended source frequency weighting by placing the proper filter in the Fourier transform or focal plane. The exact correction filter may not be physically realizable, as is the case for the $\sin(x)/x$ frequency distortion. However, we can make a good approximation to this filter and still enhance the shadow images. An approximation to the inverse filter for $\sin(x)/x$ is shown in Figure 3-b. Figure 4 shows the actual photographic filter used in our experiments. No phase shift compensation is included in the filter and the transmission range extends over a physically realizable range of zero to 100% instead of the idealized zero to infinite transmission range. Figure 4 also shows two exposures of the Fourier transform of a line-source shadow image. The $|x/\sin(x)|$ filter was designed to correct for this line-source distortion. Filter regions of 100% transmission correspond to the severely attenuated frequency regions of the object.



FIGURE 4. (a) & (b) SHOW THE AMPLITUDE OF THE FOURIER TRANSFORM OF A TEST PATTERN BLURRED BY SHADOW IMAGING WITH A LINE SOURCE. (c) SHOWS THE AMPLITUDE OF THE FILTER FUNCTION USED TO DEBLUR THE TEST PATTERN IMAGE.

The filter approximating the idealized line extended-source correction filter produced improved image quality (Figure 5-b). Since the filter was one-dimensional, edge sharpening appears only for vertical lines. As indicated, the greatest edge sharpening and contrast enhancement occurred at the higher spatial frequencies, which correspond to the highly attenuated frequencies of $\sin(x)/x$. Note that a significant sharpening occurs in all vertical lines.

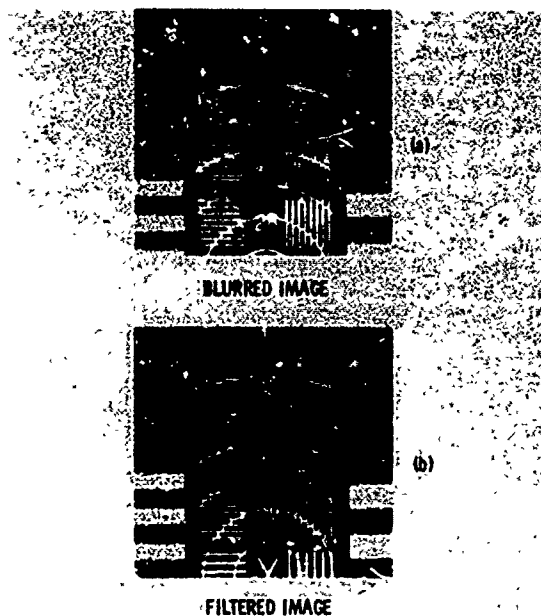


FIGURE 5. (a) IS A PHOTOGRAPH OF THE TEST PATTERN AS IMAGED BY AN EXTENDED LINE SOURCE. (b) SHOW THIS TEST PATTERN AFTER SHARPENING IN THE OPTICAL PROCESSOR.

Walsh Filtering of Extended Source Shadow Images

The initial problem in digital enhancement of shadow images is to evaluate the Walsh coefficients [5,6] in the Walsh series representation of $x/\sin(x)$. These coefficients are related to the thickness of plates containing Walsh function patterns which when assembled would provide the attenuation proportional to $x/\sin(x)$ yielding the needed optical filter. The development of $\sin(x)/x$ in a Walsh series offered no difficulty. The results are indicated in Figure 6 where increasing the number of terms gives a closer approximation. However, the function is discontinuous beyond the range of the Walsh function definition, $0 \leq x < 1$. Consequently, αx replaced x in the development, α being used to obtain the needed orders. Even though the program converged rapidly, Figure 7 shows that greater precision is needed for large c (>20) for convergence when x is near 1. Figure 7 also shows that the inclusion of higher frequency terms yields stability for greater values of x .

The evaluation of $x/\sin(x)$ offered serious problems in that the function is not square integrable. Anticipating this difficulty, a series expansion was developed for the integral of $x/\sin(x)$ using Cauchy principle values when the limits spanned poles of the integrand. The subsequent series was found to have two limit points. Using three iterations of Cesaro

summability, we found poor convergence. This difficulty was overcome by analytically extracting the Cesaro summable part and substituting its limit. Additional speed in summation was had by extracting another summable piece and substituting its limit. The remaining series converged inversely with the fourth power of the summation index. The resulting series representation of $\alpha x/\sin(\alpha x)$ is illustrated in Figure 8 for $\alpha = 10$. Figure 9 shows $\alpha x/\sin(\alpha x)$ for $\alpha = 20$. Note that the type of instability encountered in $\sin(20x)/20x$ (Figure 7) does not occur.

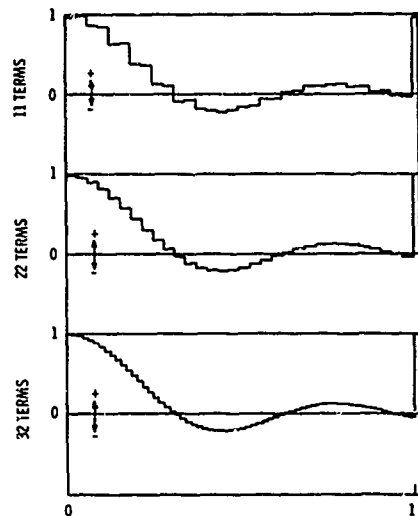


FIGURE 6. WALSH SERIES EXPANSION OF $\sin(10x)/10x$

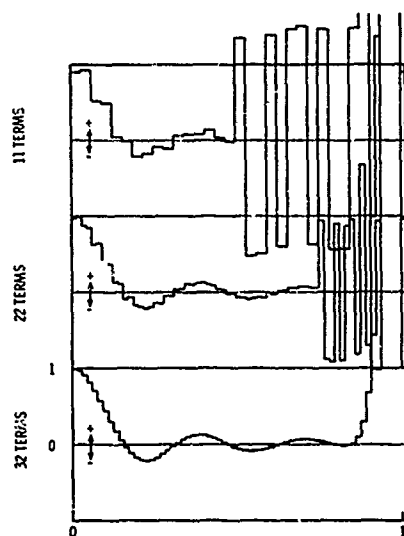


FIGURE 7. WALSH FUNCTION EXPANSION OF $\sin(20x)/20x$

Conclusions

The Walsh coefficients give the attenuation values needed for each Walsh pattern. The superposition of these patterns will then

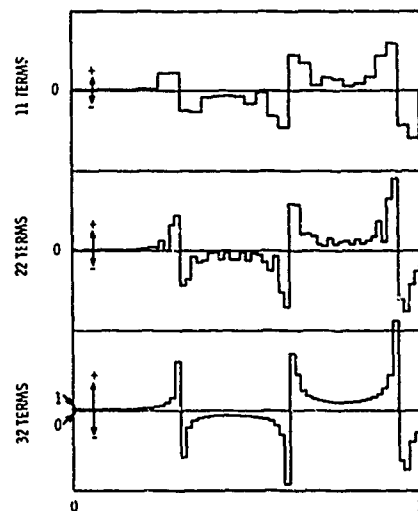


FIGURE 8. WALSH SERIES EXPANSION OF $10x/\sin(10x)$

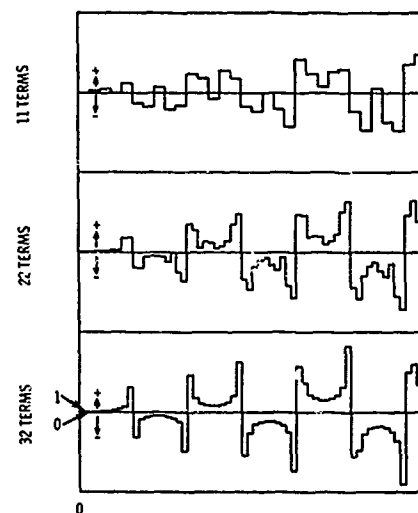


FIGURE 9. WALSH SERIES EXPANSION OF $20x/\sin(20x)$

modulate the light according to $\alpha x/\sin(\alpha x)$ which will mimic the process used in optical filtering. However, the construction of Walsh masks is costly and appropriate for only one filter. A more general treatment of the problem is needed. The development of algorithms which perform the counterpart of optical filtering within the computer would allow flexibility in the construction of filters for a wider class of image enhancement problems. Although we have demonstrated the fitting to arbitrary precision of a filter function with Walsh series, the limitations of this technique are unknown. Further, Walsh transforms of convoluted images may not yield products of the transforms of the images as does Fourier integrals of convoluted images. A deeper investigation of these problems is essential to the successful digital computation of filter functions and computerized image enhancement.

ACKNOWLEDGEMENTS

The authors express their appreciation for the suggestions of Dr. A. G. Gibbs which led to the successful summation of the series representations for the Walsh coefficients and to Mr. V. L. Crow for his help in the creation of computerized plotting programs.

REFERENCES

- [1] D. A. Ansley, "Photo Enhancement by Spatial Filtering," *Electro-Optical Systems Design* July/August 1969.
- [2] L. D. Harmon and R. C. Knowlton, "Picture Processing by Digital Computer," *Science* 164 (1969).
- [3] E. O. Brigham and R. E. Morrow, "The Fast Fourier Transform," *IEEE Spectrum* 4:63 (1967).
- [4] W. I. Pratt, J. Kane and H. C. Andrews, "Hadamard Transform Coding," *Proc. IEEE* 57:1 (1969).
- [5] N. J. Fine, "On the Walsh Functions," *Trans. Am. Math. Soc.*, V. 65, pp. 372-415, 1949.
- [6] G. W. Morgenthaler, "On Walsh-Fourier Series," *Trans. Am. Math. Soc.* V. 84, pp. 472-507, 1947.

APPENDIX

Analytical development of Walsh series Expansions

A. The evaluation a_r and b_r in the Walsh series representation of

$$\frac{\sin \alpha x}{\alpha x} = \sum_{r=0}^{\infty} a_r \text{Cal}(r, x) + b_r \text{Sal}(r, x) \quad (i).$$

where $0 \leq x < 1$, $\alpha > 0$.

Taking the product of (i) with $\text{Cal}(s, x)$ and integrating over x ,

$$a_s = \int_0^1 \frac{\sin \alpha x}{\alpha x} \text{Cal}(s, x) dx, \quad s = 0, 1, 2, \dots \quad (ii)$$

In (ii), use has been made of the orthogonality conditions for Walsh functions:

$$\int_0^1 \text{Cal}(r, x) \text{Cal}(s, x) dx = \delta_{rs}$$

$$\int_0^1 \text{Sal}(r, x) \text{Cal}(s, x) dx = 0$$

Taking the product of (i) with $\text{Sal}(s, x)$ and integrating over x ,

$$b_s = \int_0^1 \frac{\sin \alpha x}{\alpha x} \text{Sal}(s, x) dx, \quad s = 0, 1, 2, \dots \quad (iii)$$

In (iii) the additional orthogonality condition

$$\int_0^1 \text{Sal}(r, x) \text{Sal}(s, x) dx = \delta_{rs}$$

has been used.

B. The evaluation of a_r and b_r in the Walsh series representation of

$$\frac{\alpha x}{\sin \alpha x} = \sum_{s=0}^{\infty} a_r \text{Cal}(r, x) + b_r \text{Sal}(r, x) \quad (iv)$$

where $0 \leq x < 1$, $\alpha > 0$.

The same steps used in A yielded:

$$a_s = \int_0^1 \frac{\alpha x}{\sin \alpha x} \text{Cal}(s, x) dx, \quad s = 0, 1, 2, \dots \quad (v)$$

$$b_s = \int_0^1 \frac{\alpha x}{\sin \alpha x} \text{Sal}(s, x) dx, \quad s = 0, 1, 2, \dots \quad (vi)$$

C. The partition of integrals of Walsh function.

The integrals to be evaluated in (ii), (iii), (v), and (vi) are of the form

$$\epsilon_s = \sum_{k=0}^{N(s)} (-1)^k \int_{x_k(s)}^{x_{k+1}(s)} g(x) dx, \quad (vii)$$

where the $\{x_k(s)\}$ are the x 's for which the Walsh function are discontinuous. This set of x 's is evaluated from the given sequence s in a sequence of arithmetic operations. The number of values (+1 or -1) the given function takes for $0 \leq x < 1$ is evaluated.

$$k = \begin{cases} 2s + 1 & \text{for Cal}(s, x) \\ 2s & \text{for Sal}(s, x) \end{cases} \quad (viii)$$

The Walsh function index is evaluated from k by the following:

I Let $k_2 = k - 1$;

II Let $k_2 = \sum_{i=0}^{\infty} C_i 2^i$ where $C_i \in \{0, 1\}$

III The Walsh function index is N where

$$N = k_2 \div [k_2/2] = \sum_{i=0}^{\infty} c_i 2^i \div \sum_{i=0}^{\infty} c_i 2^{i-1} \\ = \sum_{i=0}^{\infty} (c_i \div c_{i+1}) 2^i = \sum_{i=0}^{\infty} d_i 2^i$$

In III, $d_i = \begin{cases} 0 & \text{if } c_i = c_{i+1} \\ 1 & \text{otherwise} \end{cases}$, this defines

dyadic addition (\div). The non-zero d_i in the evaluation of N have indices i which denote the Rademacher functions which are factors of the Walsh function $W_N(x)$ which corresponds to the given function having k constant values. The value of each factor Rademacher function is found with the following algorithm:

Given the set of i 's for which $d_i \neq 0$,

IV Let $M = [2^{i+1}x]$ where $[]$ denotes the largest integer contained in.

V If $\begin{cases} M \text{ is even } R_i(x) = 1 \\ M \text{ is odd } R_i(x) = -1 \end{cases}$.

The value of $W_N(x)$ is found from the product of the values of the $R_i(x)$ for which $d_i \neq 0$.

However, we seek the set of x 's at which $W_N(x)$ is discontinuous. These are evaluated by determining the maximum number of intervals from the $d_i \neq 0$.

VI Let $M = \max(i \in d_i)$, then $W_N(x)$ contains no more than 2^{M+1} values.

These are $\{x_{Mj}\} = \{j/2^{M+1}\}$, $j = 0, 1, 2, \dots, 2^{M+1}$

let $\{x_{ij}\} = \{j/2^{i+1}\}$, $i \leq i$ taken from the $d_i \neq 0$,

$$j = 0, 1, 2, \dots, 2^{i+1}$$

Ignoring the first entry in each set, the surviving set of x_{ij} 's are those elements of

$\{\{x_{ij}\}\}; i \in \{i \neq 0\}$ which have an odd number of occurrences for fixed i .

D. Evaluation of the integrals.

The evaluation of a_s or b_s proceeds with the evaluation of the integrals in (ii), (iii), (v), and (vi). The appropriate coefficient may be chosen for ξ_s and the appropriate function for $g(x)$ in (vii). Denoting

$$\int_{x_j}^{x_{j+1}} g(x) dx = G(x_{j+1}) - G(x_j), \text{ (vii) becomes:}$$

$$\xi_s = \sum_{j=0}^{N(s)} (-1)^j \{G(x_{j+1}) - G(x_j)\}, x_0 = 0 \text{ and } x_{N+1} = 1,$$

$$= -G(0) + 2 \sum_{j=1}^{N-1} (-1)^{j+1} G(x_j) + (-1)^N G(1) \quad (ix)$$

The function in (i) and (iii) is the Sin integral. The values of $G(x)$ are evaluated through the use of an infinite series expansion of $\text{Sin}(\alpha x)$. Here,

$$G(v) = \int^v \frac{\text{Sin}(\alpha x)}{\alpha x} dx = \frac{1}{\alpha} \sum_{k=0}^{\infty} (-1)^k \frac{(\alpha x)^{2k+1}}{(2k+1)(2k+1)!}$$

(x)

The function in (v) and (vi) is not defined. The values of $G(x)$ are determined through the following procedure:

$$\text{Here } G(v) = \int^v \frac{\alpha x}{\text{Sin} \alpha x} dx = \frac{1}{\alpha} \int^{\alpha v} \frac{Z}{\text{Sin} Z} dZ \quad (xi)$$

We can obtain a Mittag-Leffler expansion of the integrand in terms of its simple isolated poles.

$$\frac{Z}{\text{Sin} Z} = \sum_{k=-\infty}^{\infty} \frac{C_k}{(Z - k\pi)} \quad (xii)$$

$$\text{where } C_k = \lim_{Z \rightarrow k\pi} (Z - k\pi) Z / \text{Sin} Z = (-1)^k k\pi.$$

Suppose we seek a definite integral of (xii) over an interval containing a pole:

$$a \leq Z \leq u, a < k\pi < u$$

$$\text{Let } I = \int_a^u \frac{Z}{\text{Sin} Z} dZ = \sum_{\substack{k \neq 0 \\ k=-\infty}}^{\infty} (-1)^k k\pi \int_a^u \frac{dZ}{(Z - k\pi)}$$

$$= \sum_{\substack{k \neq 0 \\ k=-\infty}}^{k-1} (-1)^k k\pi \int_a^u \frac{dZ}{(Z - k\pi)}$$

$$\begin{aligned}
& + \sum_{\substack{k \neq 0 \\ k=l+1}}^{\infty} (-1)^k k \pi \int_a^u \frac{dz}{(z-k\pi)} \\
& + \lim_{\epsilon \rightarrow 0} (-1)^l l \pi \left\{ \int_a^{l\pi-\epsilon} \frac{dz}{z-l\pi} + \int_{l\pi+\epsilon}^u \frac{dz}{z-l\pi} \right\} \\
I = & \sum_{\substack{k \neq 0 \\ k=-\infty}}^{l-1} (-1)^k k \pi \left\{ \ln(u-k\pi) - \ln(a-k\pi) \right\} \\
& + \sum_{\substack{k \neq 0 \\ k=l+1}}^{\infty} (-1)^k k \pi \left\{ \ln(k\pi-u) - \ln(k\pi-a) \right\} \\
& + \lim_{\epsilon \rightarrow 0} (-1)^l l \pi \left\{ \ln \left(\frac{l\pi - (l\pi - \epsilon)}{l\pi - a} \right) \right. \\
& \left. \left(\frac{u - l\pi}{l\pi + \epsilon - l\pi} \right) \right\} \\
= & \sum_{\substack{k \neq 0 \\ k=-\infty}}^{\infty} (-1)^k k \pi \ln|u - k\pi| - \sum_{\substack{k \neq 0 \\ k=-\infty}}^{\infty} (-1)^k k \pi \ln|a - k\pi|
\end{aligned}$$

Since I has the form $G(u) - G(a)$, we may choose to evaluate (xi) as follows:

$$G(v) = \frac{1}{\alpha} \sum_{\substack{k \neq 0 \\ k=-\infty}}^{\infty} (-1)^k k \pi \ln|\alpha v - k\pi| \quad (\text{xiii})$$

Equation (xiii) may be put in the form

$$G(v) = \frac{\pi}{\alpha} \sum_{k=1}^{\infty} (-1)^k k \ln \left| \frac{1 - \alpha v/k\pi}{1 + \alpha v/k\pi} \right| \quad (\text{xiv})$$

In (xiv) convergence is still slow. Using the series expansion

$$\ln \left| \frac{1-x}{1+x} \right| = -2 \sum_{k=1}^{\infty} x^{2k-1} / (2k-1)$$

for $|x| < 1$ as a model, we may add and subtract the term $2\alpha v/k\pi$ within the summation.

$$\begin{aligned}
G(v) = & \frac{\pi}{\alpha} \sum_{k=1}^{\infty} (-1)^k k \left\{ \ln \left| \frac{1 - \alpha v/k\pi}{1 + \alpha v/k\pi} \right| \right. \\
& \left. + 2\alpha v/k\pi - 2\alpha v/k\pi \right\}
\end{aligned}$$

taking out the second sum

$$\begin{aligned}
G(v) = & \frac{\pi}{\alpha} \sum_{k=1}^{\infty} (-1)^k k (2\alpha v/k\pi) + \frac{\pi}{\alpha} \sum_{k=1}^{\infty} (-1)^k \\
& \cdot k \pi \left\{ \ln \left| \frac{1 - \alpha v/k\pi}{1 + \alpha v/k\pi} \right| + 2\alpha v/k\pi \right\} \\
= & \alpha v \sum_{k=1}^{\infty} (-1)^{k+1} = \frac{\pi}{\alpha} \sum_{k=1}^{\infty} (-1)^k \\
& \cdot k \left\{ \ln \left| \frac{1 - \alpha v/k\pi}{1 + \alpha v/k\pi} \right| + 2\alpha v/k\pi \right\}.
\end{aligned}$$

The first series has partial sum $(1, 0, 1, 0, 1, \dots)$. The n th Cesaro sum

$G_n = (n/2)/n + 1/2$ as $n \rightarrow \infty$. Replacing the sum by its Cesaro limit, $G(v) \sim v$

$$\begin{aligned}
& + \frac{\pi}{\alpha} \sum_{k=1}^{\infty} (-1)^k k \left\{ \ln \left| \frac{1 - \alpha v/k\pi}{1 + \alpha v/k\pi} \right| \right. \\
& \left. + 2\alpha v/k\pi \right\}. \quad (\text{xv})
\end{aligned}$$

However, (xv) still converges like $1/k^2$ for large k and is too slow for our purpose. We may extract another term by the above method. This term is $2(\alpha v/k\pi)^3/3$.

$$\begin{aligned}
G(v) = & v + \frac{\pi}{\alpha} \sum_{k=1}^{\infty} (-1)^k \\
& \cdot k \left\{ \ln \left| \frac{1 - \alpha v/k\pi}{1 + \alpha v/k\pi} \right| \right. \\
& \left. + 2\alpha v/k\pi + 2(\alpha v/k\pi)^3/3 - 2(\alpha v/k\pi)^3/3 \right\} \\
= & \frac{\pi}{\alpha} \sum_{k=1}^{\infty} (-1)^k k 2(\alpha v/k\pi)^3/3 + v \\
& + \frac{\pi}{\alpha} \sum_{k=1}^{\infty} (-1)^k k \left\{ \ln \left| \frac{1 - \alpha v/k\pi}{1 + \alpha v/k\pi} \right| \right. \\
& \left. + \alpha v/k\pi + 2\alpha v/k\pi + 2(\alpha v/k\pi)^3/3 \right\}.
\end{aligned}$$

The initial sum reduces to

$$\frac{2}{3} \frac{\alpha^2 v^3}{2} \sum_{k=1}^{\infty} (-1)^{k+1} / k^2 + \alpha^2 v^3 / 18. \text{ Hence}$$

$$G(v) \sim v + \alpha^2 v^3 / 18 + \frac{\pi}{\alpha} \sum_{k=1}^{\infty} (-1)^k$$

$$\cdot k \left\{ \ln \left| \frac{1 - \alpha v / k\pi}{1 + \alpha v / k\pi} \right| + 2\alpha v / k\pi + 2(\alpha v / k\pi)^3 / 3 \right\}. \quad (\text{xvi})$$

The sum converges as $1/k^4$ and yields adequate numerical performance.

Consequences of the foregoing expansions.

Consider the product

$$\frac{\sin \alpha x}{\alpha x} \cdot \frac{\alpha x}{\sin \alpha x} = \left(\sum_{k=0}^{\infty} C_k W_k(x) \right) \left(\sum_{\ell=0}^{\infty} D_{\ell} W_{\ell}(x) \right) \quad (\text{xvii})$$

for x, α in $I : 0 \leq x < 1, \alpha > 0$.

Let $m = k * \ell$, then we can evaluate $k = m * \ell$. Here, the symbol $*$ refers to dyadic difference: and is defined to yield the sets $\{m\}$, $\{k\}$, and $\{\ell\}$ which obey the first equation. We make use of the relation $W_k(x)W_{\ell}(x) = W_m(x)$ and write (xvii) equivalent to its reduced left hand side.

$$\sum_{m=0}^{\infty} W_m(x) \sum_{\ell=0}^m C_{m * \ell} D_{\ell} = 1; x, \alpha \text{ in } I. \quad (\text{xviii})$$

Since the Walsh functions $\{W_m(x)\}$ form an orthonormal basis on I ,

$$\sum_{\ell=0}^m C_{m * \ell} D_{\ell} = \begin{cases} 1 & \text{if } m=0 \\ 0 & \text{otherwise} \end{cases}. \quad (\text{xix})$$

We may now use (xix) to evaluate the D_{ℓ} given the C_k , thus avoiding the series expansion given in (xvi).

A WIRED-IN RESISTOR CIRCUIT REALIZATION OF THE TWO-DIMENSIONAL HADAMARD-TRANSFORMATION OF BROADBAND TELEVISION SIGNALS

UWE KRAUS

INSTITUT FOR TECHNISCHE ELEKTRONIK DER RWTH AACHEN, GERMANY

Introduction

In recent years the transmission of broadband TV-signals by PCM-system could be realized. In PCM-technique signal regeneration without decrease of the S/N-ratio can easily be done. The disadvantage is the great demand for frequency band-width.

Therefore, in many papers (1) suggestions have been discussed to reduce TV-signal band-width. All these techniques try to decrease the bit-rate by redundancy reduction and a suitable reduction of irrelevant information.

The transformation method is one of these techniques. An orthogonal transformation is applied to the sampled picture elements. The coefficients are decorrelated and represent the redundancy reduced picture information. A reduction of irrelevant information is possible by spatial filtering. This can be done by suppression of some coefficients and quantization of the other. The picture is reconstructed by transforming back into the original domain.

The transformation by means of the Hadamard matrix can easily be performed because the elements of the matrix have only the values +1 and -1. The results of computer simulations have shown that a high data compression can be achieved especially by the use of the two-dimensional Hadamard transformation.

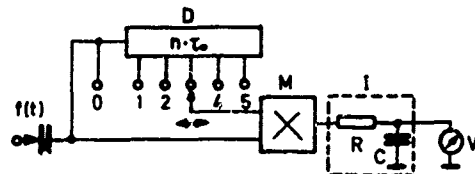
The matrix multiplication of the Hadamard transformation can be performed on a digital computer as well as by means of an analogous technique using resistor and amplifier networks. This will now be discussed with respect to real time operation of moving pictures in future.

At first, an experimental set is described that transforms TV-pictures scanned in CCIR-standard in sub-frames into the two-dimensional Hadamard domain and back again. After this some considerations of possibilities of realization of real time operation are made.

Experimental set for performing the two-dimensional Hadamard Transformation in sub-frames

Size of the sub-frames

The size of the adjacent sub-frames was fixed, so that on an average for a lot of TV-pictures only a few correlation should exist between picture elements lying on opposite edges.



D: Delay-line, $\tau_0 = 0.1 \mu s$, M: Multiplier, I: Integrator, $RC = 1s$, V: Voltmeter

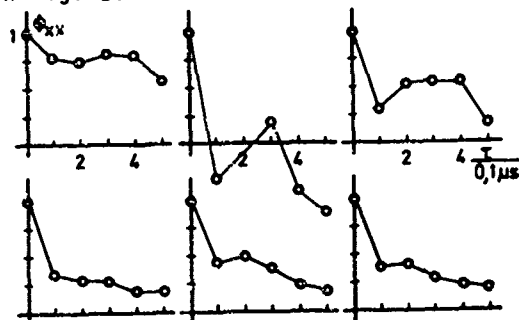
Fig. 1 Block diagram of auto-correlator

By means of the experimental set shown in Fig. 1 the autocorrelation functions of a lot of diapositives scanned in CCIR-standard are measured. The set performs the function

$$\Phi_{xx}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_0^T f(t) \cdot f(t-\tau) dt \quad (1)$$

The delay-line is adjustable to steps of $0.1 \mu s$. The integration is carried out with a time-constant of $1s$, so the average value of many TV-frames is achieved.

Some of the measured functions are shown in Fig. 2.



One can see that the rather different looking curves still have a relative big value at $\tau = 0,5 \mu s$. Therefore a sub-frame size of 8×8 picture elements was provided. But because of costs a sub-frame size of 4×4 picture elements was realized.

Principle of operation of the experimental set

Scanning scheme

To perform the transformation it is necessary that the samples of the sub-frame to be transformed are all present at the same time. As the picture elements are scanned sequentially line by line, the elements belonging to one sub-frame are stored in sample and hold circuits.

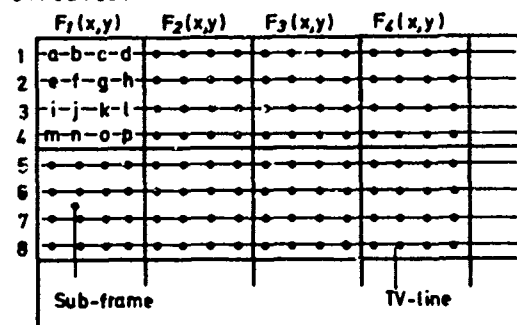


Fig. 3 Division of the TV-frame into sub-frames

Fig. 3 shows the partition of the TV-frame into sub-frames. During the first TV-line the picture elements a, b, c, d of the sub-frame $F_1(x,y)$ are stored each in one sample and hold circuit. During the next TV-line - this is the third because of interlace system - the picture elements i, j, k, l are stored. The elements of the second and fourth line must be stored on succession of the next frame. After having transformed the stored picture elements the elements of sub-frame $F_2(x,y)$ are stored and so on.

The whole TV-picture is subdivided into 168 sub-frames in horizontal and 145 in vertical direction.

Block diagram of the set

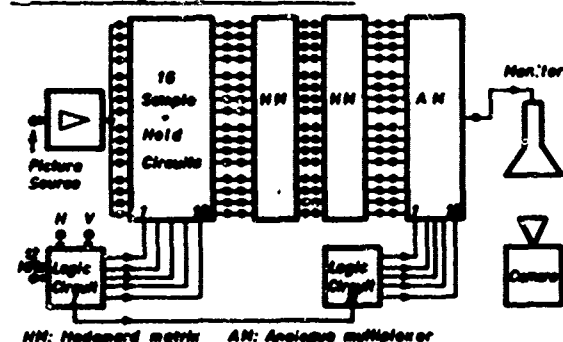


Fig. 4 Block diagram of experimental set

Fig. 4 shows the block diagram of the whole set. An amplifier feeds the analogous video signal of the picture source (e.g. a flying spot scanner) simultaneously in the inputs of 16 sample and hold circuits. The logic circuit I derives 16 sequential triggerpulses from the 12 MHz master clock and the synchronizing pulses H and V. The trigger pulses open the sample and hold circuits for a time of $0,08 \mu s$ using the scheme shown in Fig. 3.

A matrix circuit HM transforms the stored signals into the Hadamard domain. A second circuit of the same kind performs the transformation back into the original domain. Short samples of $0,08 \mu s$ duration are taken sequentially from the retransformed signals by means of an analogue multiplexer and led to a TV-monitor which is also synchronized by the pulses H and V.

The logic circuit II controls the multiplexer in a way that the retransformed picture elements are written on the monitor screen in the same succession as the original elements. The reproduced sub-frame appears on the screen in the original position. The rest of the screen is dark.

Storing and reproducing of one sub-frame takes a time of four TV-frames. All sub-frames are processed in half an hour. As the monitor screen is photographed during this time the complete reconstructed picture is stored on the film.

The spatial filtering can be done between the matrix circuits by suppression, thresholding and quantization.

Realization of the matrix circuit

To perform the transformation of the sub-frame into the two dimensional Hadamard domain the matrix circuit has to realize equation (2).

$$[A(k,m)] = [H(k,m)][F(x,y)][H(k,m)] \quad (2)$$

with

$A(k,m)$: matrix of coefficients

$H(k,m)$: Hadamard matrix

$F(x,y)$: matrix of picture elements of a sub-frame

in detail

$$\begin{bmatrix} A_{00} & A_{10} & A_{20} & A_{30} \\ A_{01} & A_{11} & A_{21} & A_{31} \\ A_{02} & A_{12} & A_{22} & A_{32} \\ A_{03} & A_{13} & A_{23} & A_{33} \end{bmatrix} = \begin{bmatrix} + & + & + & + \\ + & - & - & - \\ + & - & - & - \\ + & - & - & - \end{bmatrix} \cdot \underbrace{\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix}}_{[S(u,v)]} \begin{bmatrix} + & + & + & + \\ + & - & - & - \\ + & - & - & - \\ + & - & - & - \end{bmatrix} \quad (3)$$

The transformation is performed in two steps.

At first the elements of the matrix $[S(u,v)]$ are defined.

$$\begin{bmatrix} S_{00} & S_{01} & S_{02} & S_{03} \\ S_{10} & S_{11} & S_{12} & S_{13} \\ S_{20} & S_{21} & S_{22} & S_{23} \\ S_{30} & S_{31} & S_{32} & S_{33} \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} \cdot \begin{bmatrix} + & + & + & + \\ + & - & - & - \\ + & + & - & - \\ + & + & - & + \end{bmatrix} \quad (4)$$

The transformation circuit shown in Fig. 5 works with the principle of current addition. The signal voltages e.g. representing the picture elements a, b, c, d of the first row of the sub-frame control the current sources consisting of the transistors T1, T3, T5, T7 and their emitter resistors R. The collector currents are added in another transistor T9 the collector voltage of which represents the coefficient S₀₀ with negative sign. The circuit is completed by the transistors T2, T4, T6, T8, T10, and T11. The transistor pairs T1 T2, T3 T4, T5 T6, and T7 T8 represent differential amplifiers fed by the common constant current source T11. The collector voltage of T10 represents the coefficient +S₀₀.

A signal voltage fed into the base of an even numbered transistor of a differential amplifier with the base of the odd numbered transistor is grounded makes a negative contribution to the sum. In this way the signal voltages can be valued according to the sign of Hadamard matrix elements.

In the second step of transformation the equation (5)

$$[A(k,m)] = [H(k,m)][S(u,v)] \quad (5)$$

is carried out by the right part of the circuit diagram shown in Fig. 5. The transistors T12 to T27 and their emitter resistors are current sources controlled by the coefficient voltages +S_{0v} or -S_{0v} according to the Hadamard coefficients. Changes of signal polarity in the transistors T28 to T31 must be taken into consideration. The collector voltages of these transistors represent the coefficients A_{0m}.

The whole transformation circuit is built on four printed cards each of which produces four column coefficients of the matrix [A(k,m)]. As the transformation circuit consists of semiconductors and resistors only an integrated technique realization seems to be possible.

Another four cards built and wired as described transform the coefficients back into the original domain. Equation (6) is realized.

$$[F(x,y)] = \frac{1}{N^2} [H(k,m)][A(k,m)][H(k,m)] \quad (6)$$

with N: order of matrix

The gains of the matrix circuits are adjusted so that the reconstructed signals have the same value of voltage like the input signals.

Possibilities for real time-operation

Further considerations have led to conceptions of transmission of moving TV-pictures in the two-dimensional Hadamard domain. But using the usual line by line scanning scheme the samples of a sub-frame can be present at the same time not before the second TV-frame, as shown.

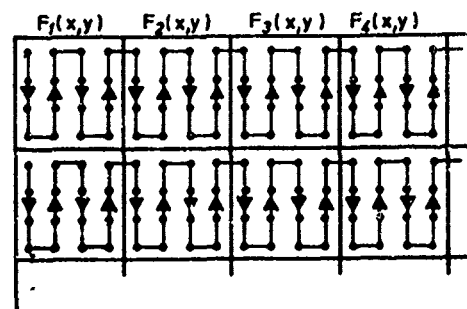


Fig. 6 Scanning scheme for real-time operation

This difficulty can be avoided by employing the modified scanning scheme of Fig. 6. Each sub-frame is scanned column by column. All sub-frames can be scanned within the time of 40 ms using the usual electron beam velocity.

Fig. 7 illustrates the block diagram of a transform apparatus working in real time. The picture source e.g. running an electrostatically deflected vidicon uses the scanning scheme shown in Fig. 6. The transformation circuit consists of two identical parts I and II working in parallel. Each part possesses an array of 16 sample and hold circuits, a transformation circuit shown in Fig. 5, and a parallel-serial-converter. A synchron pulse generator controls the set in order that e.g. the sub-frames F₁(x,y), F₃(x,y) F_{2n-1}(x,y) are processed in the circuit part I and the sub-frames F₂(x,y), F₄(x,y) F_{2n}(x,y) in part II.

part I	: store F _{2n-1} (x,y)	transmit A _{2n-1} (k,m)
part II	: transm. A _{2n} (k,m)	store F _{2n} (x,y)
position of switch S:	II	I

The coefficients are transmitted in PCM-system.

Sub-frame transformation in real-time can be carried out in digital technique too (Fig. 8). The picture source also works with the scanning scheme of Fig. 6. First of all the analogous video signal is converted to straight PCM. The transformation circuit consists of two identical parts I and II again. A demultiplexer loads the code words of the first sub-frame column to four adders 1 to 4 operating in parallel, the data of the

second column to the adders 5 to 8 etc. The adders are programmed in order that successive code words are added or subtracted according to the Hadamard matrix.

The adders 1 to 16 perform the first step, the one-dimensional transformation. The adders 17 to 32 constructed and programmed in the same way compute the two-dimensional Hadamard coefficients. Realization in TTL-technique seems to be possible because in worst case an adder must add a new code word every 0,1 μ s.

Controlled by the synchron pulse generator the circuit parts I and II work together in the same way as shown in Fig. 7.

A real-time transmission of CCIR-standard scanned pictures in the two-dimensional Hadamard domain is possible if the picture elements of four TV-lines are written into a store and read out in such a sequence that picture elements belonging to a sub-frame appear successively at the store output.

Such a store can be realized by means of random access read/write memories. At first the video signal is analogue/digital converted. The code words of four TV-lines are stored with a definite succession of addresses. They are read out with a different succession of addresses so that the code words of a sub-frame appear successively. They can be processed in a digital transformation circuit as shown in Fig. 8. During the read out the picture elements of the next four TV-lines are stored in a second digital store.

Another realization of such a scanning scheme converter is possible. Consider a long digital shift register that stores all code words of four TV-lines. This shift register consists of sub-registers in series arrangement with the capacity to store the bits of one sub-frame row. After the data of four TV-lines have been read in, the shift register is reorganized by gates in order that a new long shift register is formed with the sub-registers of the sub-frames in series arrangement. Loading out the new shift register the code words leave in the wanted succession.

Conclusion

An experimental set was described that transforms TV-pictures into the two-dimensional Hadamard domain and vice versa. Compression ratio and picture quality are similar to the results of computer simulations. Further more it was shown that real-time transmission of TV-signals in the two-dimensional Hadamard domain is possible by means of analogous as well as digital circuits. A modified scanning scheme and CCIR standard can be used. Some problems e.g. dealing with interlace still must be solved. Further technical and financial considerations must decide what sort of system will be realized.

References

1. Proc. IEEE, Vol. 55 No. 3 (March 1967)
2. Harmuth, H.F.
Sequenzfilter mit zwei Raumvariablen und LCS-Filter
Nachrichtentechnische Zeitschrift 23 Heft 8 August 1970, S. 377-383
3. Pratt, W.K., Kane, J., Andrews, H.C.
Hadamard Transform Image Coding
Proc. IEEE Vol. 57 No. 1 (January 69)
4. Andrews, H.C.
Computer Techniques In Image Processing
Academic Press 1970
5. Landau, H.J., Slepian, D.
Some Computer Experiments in Picture Processing for Bandwidth Reduction
The Bell System Technical Journal Vol. 50 No. 5 May - June 1971
6. Viebahn, H.
Entwurf und Aufbau einer Apparatur zur blockweisen Realisierung einer zweidimensionalen Transformation von seriell anfallenden Abtastdaten
Diplomarbeit DA 626/Ks am Institut für Technische Elektronik der RWTH Aachen, 1971

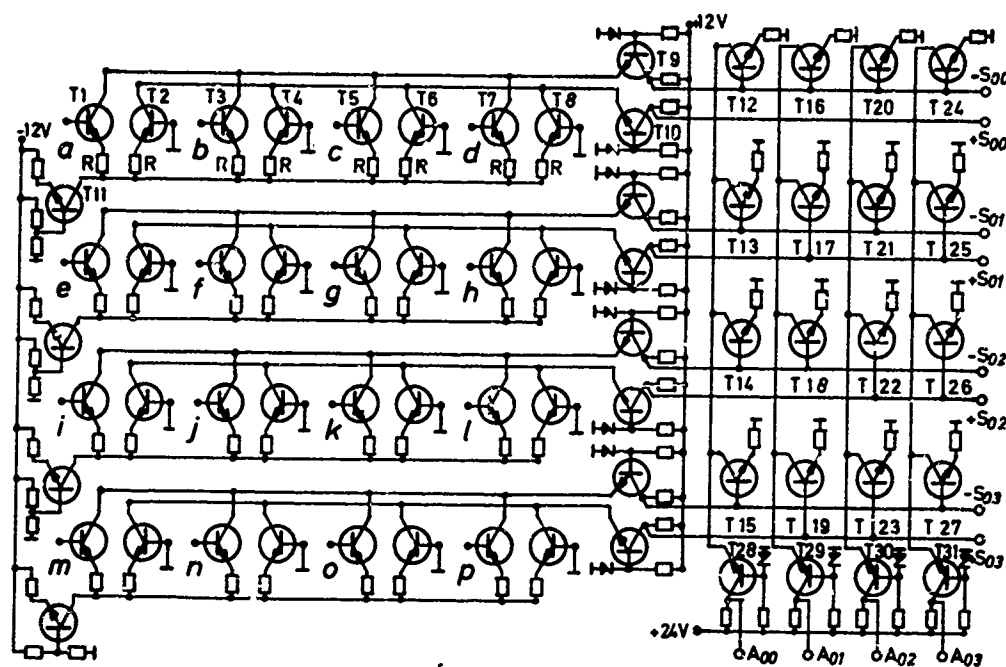
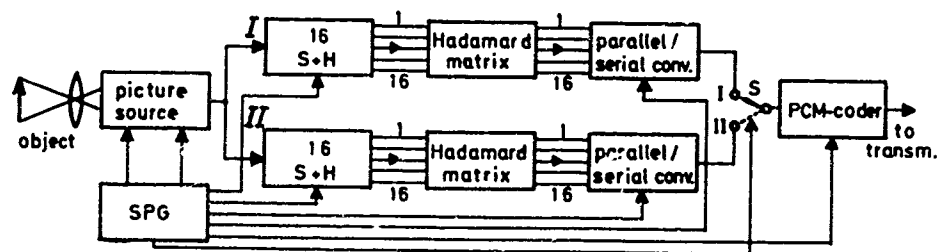


Fig. 5 Circuit diagram of transformation matrix



S-H: Sample and Hold circuit, SPG: Synchron Pulse Generator

Fig. 7 Hadamard encoder for real-time operation (analogous version)

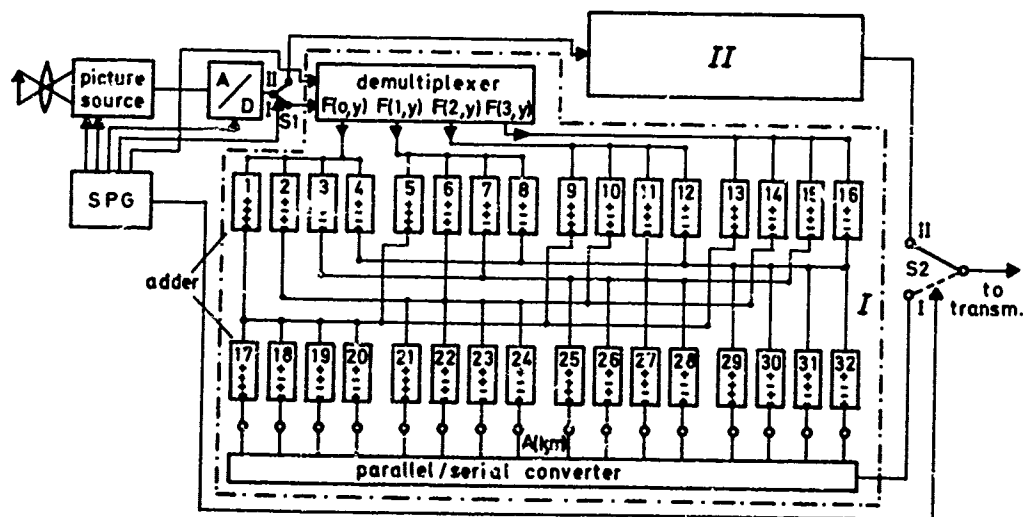


Fig. 8 Hadamard encoder for real-time operation (digital version)

SLANT TRANSFORMS FOR IMAGE CODING*

William K. Pratt

Lloyd R. Welch

Wen-hsiung Chen

Department of Electrical Engineering

University of Southern California

Los Angeles, California

Introduction

In 1968 the concept of coding and transmitting the two dimensional Fourier transform of an image, computed by a fast computational algorithm, rather than the image itself, was introduced [1,2]. This was followed shortly thereafter by the discovery that the Walsh-Hadamard transform could be utilized in place of the Fourier transform with a considerable decrease in computational requirements [3]. Investigations then began into the application of the Karhunen-Loeve [4] and the Haar [5] transforms for image coding. The Karhunen-Loeve transform provides minimum mean square error coding performance but does not possess a fast computational algorithm. On the other hand the Haar transform has the attribute of an extremely efficient computational algorithm, but results in a relatively large coding error. None of the transforms mentioned above, however, has been expressly tailored to the characteristics of an image.

A desirable property for an image coding transform is that the transform compact the image energy to as few of the transform domain samples as possible. Qualitatively speaking, a high degree of energy compaction will result if the basis vectors of the transformation matrix "resemble" typical horizontal or vertical lines of an image. If one examines the lines of a typical monochrome image, it is found that a large number of the lines are of nearly constant grey level over a considerable length. The Fourier, Hadamard, and Haar transforms possess a constant valued basis vector that provides an efficient representation for constant grey level image lines, while the Karhunen-Loeve transform has a nearly constant basis vector suitable for this representation. Another type of typical image line is the line that linearly increases or decreases in brightness over the length. None of the data transforms previously mentioned possess a basis vector that efficiently represents such image lines.

Shibata and Enomoto have introduced orthogonal transformations containing a "slant" basis vector for data of vector lengths of four and eight [6]. The slant vector is a discrete sawtooth waveform decreasing in uniform steps over its length, and is suitable for efficiently representing gradual brightness changes in an image line. Their work gives no indication of a construction for larger data vectors, nor exhibits the use of a fast computational algorithm. In order to achieve a high degree of image coding compression with transform coding techniques, it is necessary to perform the transformation in two dimensions over block sizes of 16×16 picture elements or greater [7]. For large block sizes, computation is usually not feasible unless a fast algorithm is employed.

With this background an investigation was undertaken to develop an image coding slant transform matrix possessing the following properties:

1. orthogonal set of basis vectors.
2. constant basis vector.
3. slant basis vectors.
4. sequency property.
5. variable size transformation.
6. fast computational algorithm.
7. high energy compaction.

The following sections describe the construction of the slant transformation matrix, present a fast computational algorithm for its computation, discuss its image coding performance, and provide examples of its use for coding monochrome and color images.

Slant Transform Construction

For a vector length of $N = 2$ the slant transform is identical to the Hadamard transform of order 2. Thus,

$$[S_2] = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (1)$$

*This work was supported by the Advanced Research Projects Agency of the Defense and was monitored by the Air Force Eastern Test Range under Contract No. F08606-77-C-0008.

The slant transform matrix for $N = 4$ can be written as

$$[S_4] = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ a+b & a-b & -a+b & -a-b \\ 1 & -1 & -1 & 1 \\ a-b & -2-b & a+b & a+b \end{bmatrix} \quad (2)$$

where a and b are real constants to be determined subject to the conditions that S_4 must be orthogonal and that the step size of the slant basis vector must be the same throughout its length. The step size between the first two elements of the slant vector is

$$(a+b) - (a-b) = 2b \quad (3)$$

and the step size between the second and third elements is

$$(a-b) - (-a+b) = 2a - 2b \quad (4)$$

Hence,

$$a = 2b$$

The slant matrix of order four may then be reformed as

$$[S_4] = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 3b & b & -b & -3b \\ 1 & -1 & -1 & 1 \\ b & -3b & 3b & -b \end{bmatrix} \quad (5)$$

By the orthogonality condition

$$\frac{1}{\sqrt{4}} [3b \ b \ -b \ -3b] \frac{1}{\sqrt{4}} [3b \ b \ b \ 3b]^T = 1$$

it is found that

$$b = \frac{1}{\sqrt{5}} \quad a = \frac{2}{\sqrt{5}}$$

Thus, the slant matrix becomes

$$[S_4] = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} \\ 1 & -1 & -1 & 1 \\ \frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & -\frac{1}{\sqrt{5}} \end{bmatrix} \quad (6)$$

It is easily shown that S_4 is orthonormal. Further note that S_4 possesses the sequency property; each row has an increasing number of sign reversals from 0 to 3. The fast computational property of S_4 is apparent from the matrix decomposition

$$[S_4] = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{3}{\sqrt{5}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{3}{\sqrt{5}} \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & \frac{1}{3} \\ 1 & -1 & 0 & 0 \\ 0 & 0 & \frac{1}{3} & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \end{bmatrix} \quad (7)$$

If S_4 is post multiplied by a column data vector, the first computational pass requires 4 additions, the second pass requires 2 multiplications (the two elements $1/3$) and the final pass requires 4 multiplications including the normalizing factor of $1/\sqrt{4}$. The total computational requirements are 8 adds and 6 multiples. For purposes of comparison a fourth order Hadamard transform requires 8 adds and 4 multiples. Figure 1 contains a flow chart of the computational operations for S_4 .

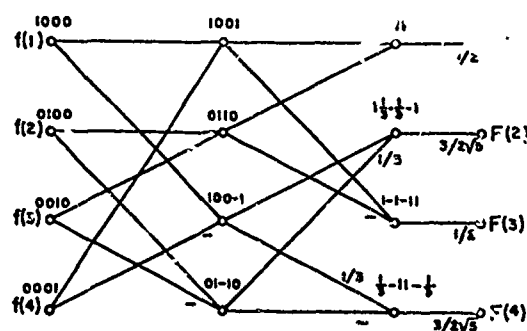


Figure 1. Slant transform of order 4-computational flowchart.

An extension of the slant matrix to its next size increment S_8 is given by

$$[S_8] = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ a_8 & b_8 & 0 & 0 & -a_8 & b_8 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ -b_8 & a_8 & 0 & 0 & b_8 & -a_8 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ \frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & \frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & -\frac{1}{\sqrt{5}} \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ \frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & \frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & -\frac{1}{\sqrt{5}} \end{bmatrix} \quad (8)$$

where a_8 and b_8 are constants to be determined to satisfy the slant and sequency properties. In S_8 the slant vector is obtained by a simple scaling operation on S_4 . The remaining terms in eq. (8) are introduced to obtain the sequency and orthogonality properties.

Equation (8) can be generalized to give the slant matrix of order N in terms of the

slant matrix of order $N/2$ by the following construction.

$$\begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} (a_{N/2}) & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \quad (2)$$

where I represents a 2×2 identity matrix. To determine the coefficients (a_N, b_N) , one proceeds as follows: The first row is a constant

$$S_N(1, i) = \frac{1}{\sqrt{N}}$$

The second row (the slant vector) is a linear function of the column index which is orthogonal to the first row. It must therefore be of the form

$$S_N(2, i) = x_N(N+1-2i)$$

Now, by the recursion indicated in eq. (9), for $i \leq N$

$$S_{2N}(2, i) = \frac{1}{\sqrt{2}} a_{2N} S_N(1, i) + \frac{1}{\sqrt{2}} b_{2N} S_N(2, i)$$

or

$$x_{2N}(2N+1-2i) = \frac{1}{\sqrt{2N}} a_{2N} + b_{2N} \frac{x_N}{\sqrt{2}} (N+1-2i)$$

From this one obtains

$$x_{2N} = b_{2N} \frac{x_N}{\sqrt{2}}$$

$$a_{2N} = \sqrt{2} N^{3/2} x_{2N}$$

and by induction

$$a_{2N} = 2b_{2N} a_N$$

Since $S_N(1, \cdot)$ and $S_N(2, \cdot)$ are orthogonal unit vectors in N dimensions and $S_{2N}(2, \cdot)$ is a unit vector in $2N$ dimensions, the above recursion implies

$$1 = \|S_{2N}(2, \cdot)\|^2 = a_{2N}^2 + b_{2N}^2$$

These two relations can be used to obtain the coefficients, (a_N, b_N) recursively:

$$a_1 = 1$$

$$b_{2N} = 1/\sqrt{1 + 4a_N^2}$$

$$a_{2N} = 2b_{2N} a_N$$

Figure 2 contains a superimposed plot of the Walsh-Hadamard and Slant basis vectors for a vector length of sixteen for the construction of eq. (9). It is interesting to note

that many of the mid-sequence basis vectors are identical for the two transforms.

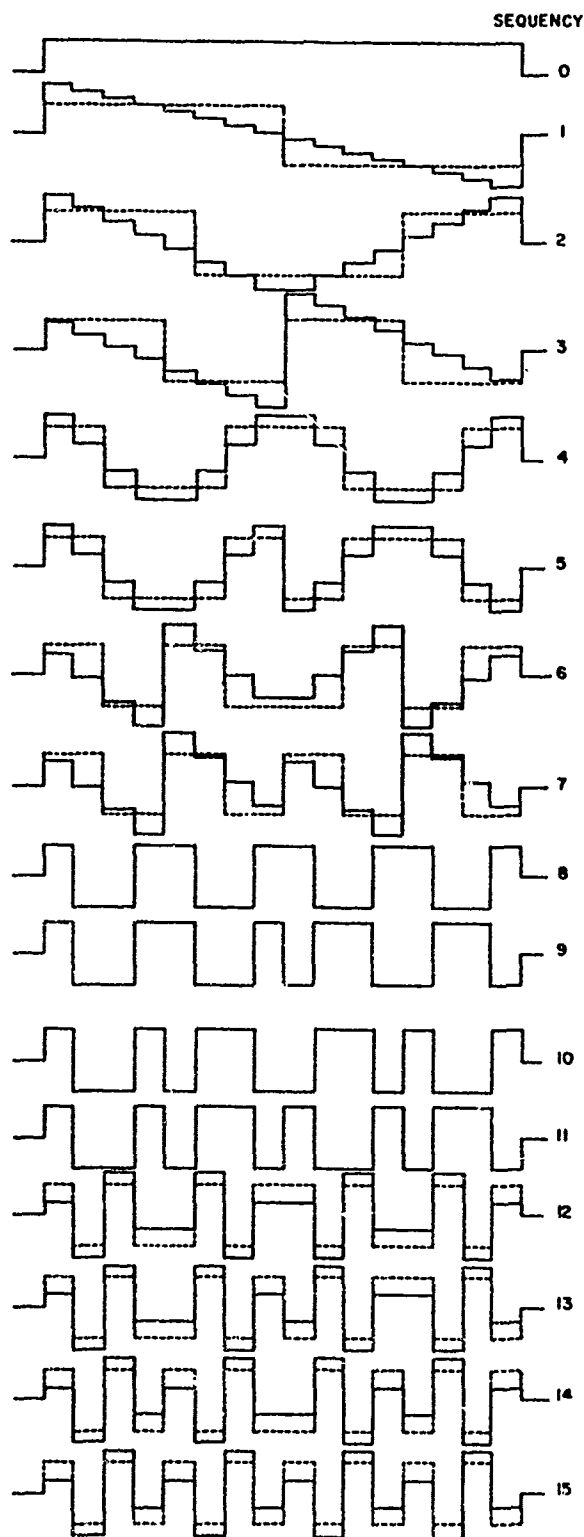


Figure 2. Comparison of Walsh-Hadamard and Slant basis vectors of length 16.

Slant Transform Properties

Let $[f]$ be a column data vector and $[F]$ its slant transform obtained by the operation

$$[F] = [S_N][f] \quad (10)$$

Consider $[f]$ to be a sample of a vector random process with known mean $[f]$ and with a known covariance matrix

$$[C_f] = \overline{([f] - [\bar{f}])([f] - [\bar{f}])^*} \quad (11)$$

where the overbar indicates a statistical average. The covariance matrix of the Slant transform $[F]$ is found to be [8]

$$[C_F] = [S_N][C_f][S_N]^T \quad (12)$$

Equation (12) can be considered a two dimensional Slant transformation of the data covariance matrix for purposes of computation. Figure 3 contains a perspective view of the



Figure 3. Perspective view of Slant transform covariance matrix-Markov process data vector, $\rho = 0.95$, $N = 256$.

Slant transform of a data vector of length $N = 256$ with a Markov process covariance of the form

$$[C_f] = \rho^{|x_i - x_j|}$$

where ρ is the correlation of adjacent elements $[f]$. Figure 4 is a plot of the variance of the Slant transform samples as a function of sequency. The variance functions for the Walsh-Hadamard, Fourier, Haar, and Karhunen-Loeve transforms are included for comparison. It is seen that the variance function for the Slant transform is reasonably close to the variance function of the Karhunen-Loeve transform, which is known to provide

the best energy compaction for the Markov source.

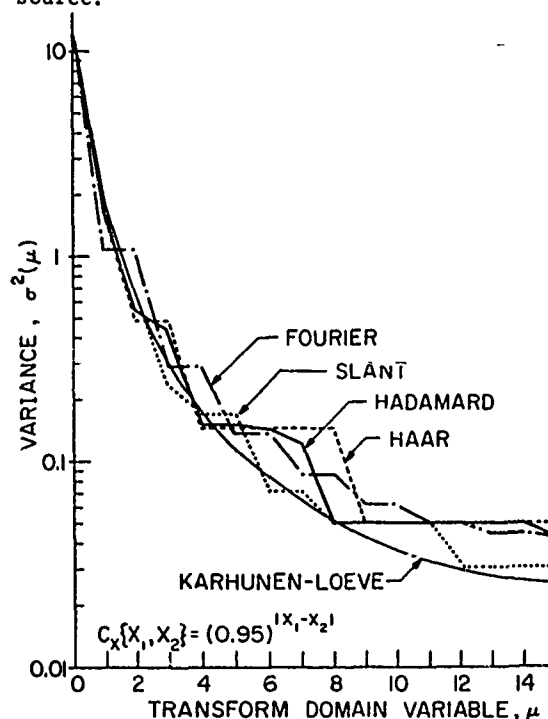


Figure 4. Transform domain variance, vector length = 16, element correlation = 0.95.

Slant Transform Image Coding

Let $[f(x, y)]$ represent the brightness samples of an N by N element image. The two dimensional Slant transform of the image is given by

$$[F(u, v)] = [S_N][f(x, y)][S_N]^T$$

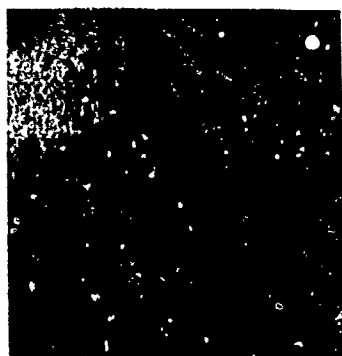
In effect, the pre-multiplication of $[f(x, y)]$ by $[S_N]$ performs a one dimensional slant transform of each column of the image matrix, and the post-multiplication by $[S_N]^T$ performs a one dimensional transform of the rows of the image. Figure 5 contains a photograph of a 256 by 256 element image with 64 grey levels and its two dimensional Slant transform.

A bandwidth reduction can be obtained with the Slant transform by efficiently quantizing each transform domain sample. There are two basic strategies for the quantization process - zonal and threshold quantization. In the former, various zones are established in the transform domain, and each sample in the zone is coded with the same number of bits set proportional to the expected variance of the samples within the zone. With threshold quantization a threshold level is established and only those transform domain samples

whose magnitude are greater than the threshold are coded.



(a)



(b)

Figure 5. Slant transform of an image: (a) original; (b) transform threshold display.

Figure 6 presents a statistical evaluation of the coding performance of the Walsh-Hadamard, Karhunen-Loeve, and Slant transforms for a form of zonal quantization in which the transform domain samples in a zone are coded with six bits per sample and samples outside the zone are discarded. The zone is defined to contain the transform domain samples with the largest expected variance, and is adjusted to include 25% of the total number of transform domain samples. Images coded with this system require an average coding of 1.5 bits per element. Figure 6 plots the mean square error resulting from this quantization process as a function of the size of the image block transformed. From the figure it is seen that the Karhunen-Loeve transform provides the minimum mean square error,

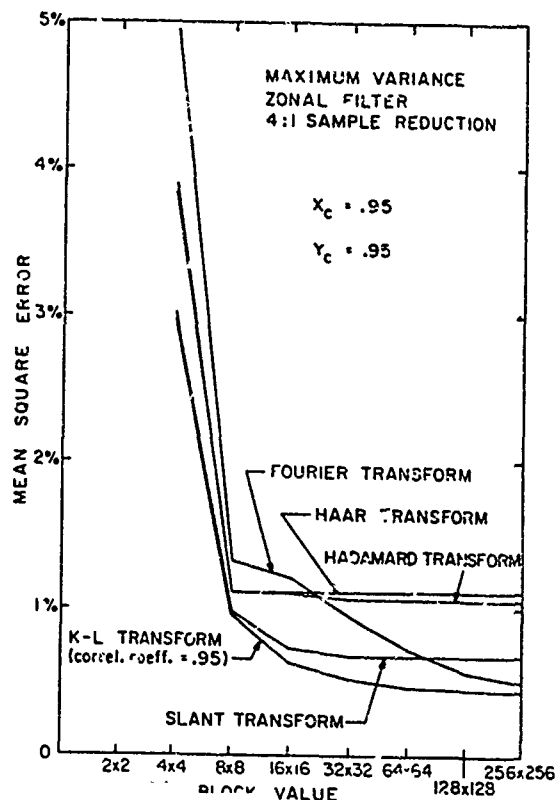


Figure 6. Mean square error performance of image transforms as a function of block size for low pass zonal quantization.

but the Slant transform results in only a slightly greater error. Also to be noted is that the rate of decrease in mean square error for larger block sizes becomes quite small after a block size of 32 by 32 elements.

Several computer simulations have been performed to evaluate the Slant transform for image coding. Figure 7 shows image reconstructions for the Walsh-Hadamard, Karhunen-Loeve, and Slant transforms for zonal quantization employing eight zones and coding with an average of only 1.5 bits per element. Subjectively, the Slant transform results in much less degradation than the Walsh-Hadamard transform and only slightly more than the Karhunen-Loeve transform. Similar experiments have been performed for color images, and it has been found that a color image can be coded with about 2.0 bits per element with

negligible degradation using the Slant transform.



(a)



(b)



(c)

Figure 7. Reduced bandwidth images obtained by transform coding with zonal quantization, 1.5 bits/pixel. (a) Walsh-Hadamard coded; (b) Slant coded; (c) Karhunen-Loeve coded.

Summary

A new orthogonal image transform with a basis vector matched to gradual brightness changes along image lines has been developed. The transform can be computed using a fast computational algorithm, and requires only a few more operations than the Walsh-Hadamard transform. A statistical analysis indicates that the Slant transform provides a smaller mean square error for image coding than the Walsh-Hadamard transform and a slightly greater error than the Karhunen-Loeve transform. The analytic image coding performance predictions are verified by computer simulations of image coding processes for monochrome and color images.

References

1. H. C. Andrews and W. K. Pratt, "Television Bandwidth Reduction by Encoding Spatial Frequencies," *Journal of the SMPTE*, Vol. 77, No. 2, December, 1968, pp. 1279-1284.
2. G. B. Anderson and T. S. Huang, "Piecewise Fourier Transformation for Picture Bandwidth Compression," *IEEE Transactions on Communication Technology*, Vol. COM-19, No. 2, April, 1971, pp. 133-140.
3. H. C. Andrews, J. Kane, and W. K. Pratt, "Hadamard Transform Image Coding," *Proceedings of the IEEE*, Vol. 57, No. 1, January, 1969.
4. A. Habibi and P. A. Wintz, "Image Coding by Linear Transformation and Block Quantization," *IEEE Transactions on Communication Technology*, Vol. COM-19, No. 1, February, 1971, pp. 50-63.
5. H. C. Andrews, *Computer Techniques in Image Processing*, Academic Press, New York, 1970.
6. H. Enomoto and K. Shibata, "Orthogonal Transform Coding System for Television Signals," *Journal of the Institute of TV Engineers of Japan*, Vol. 24, No. 2, February, 1970, pp. 99-108.
7. W. K. Pratt, "A Comparison of Digital Image Transforms," *University of Missouri at Rolla, Communications Conference*, September, 1970.
8. W. K. Pratt, "Linear and Nonlinear Filtering in the Walsh Domain," *Proceedings of the Symposium on the Application of Walsh Functions*, April, 1971, pp. 38-42.

MODIFIED TRANSFORMS IN IMAGERY ANALYSIS

G. Graham Murray
Consultant
Data/Ware Development
La Jolla, Calif. 92037

Introduction

Since detailed apprehension of the physical world depends critically upon vision, it is not surprising that investigators from many fields are currently exploring image processing. Applications include radiography, earth resources data gathered by satellites, spectrum analyzers, reconnaissance pictures, facsimile, remotely piloted vehicles, and others. The objectives are typically data compression or image enhancement. Use of the Walsh and Fourier transforms is a promising new tool, but to date this has entailed either large-scale digital computers or relatively long processing times.

Since the feasibility of these techniques has already been demonstrated, there is considerable emphasis upon obtaining practical implementations in the form of low-cost, high-speed special hardware. A previous paper (1) suggested that it was advantageous to apply 1-D (one dimensional) transforms to images. There are in existence special processors which can form the 1-D Fast Walsh and Fourier Transforms (FWT and FFT). Furthermore, a 1-D transform processor affords a better fit with current image scanning devices. Consider for instance TV sensing devices which provide a read-out one line at a time. Ideally this data would be entered into a FWT or FFT processor as it is generated and processed on-line.

In order to avoid the requirement for extremely large memories, it is convenient to partition the picture. It will be shown that applying the 1-D FWT to each of the partitions is actually equivalent to taking the 2-D FWT. In addition this can be carried out in an adaptive, iterative manner which provides insight into the mechanism of compression. The extent to which this also applies to the FFT will be indicated.

One-Dimensional FWT

There are three motivations for use of the 1-D FWT in imagery analysis:

- (1) Compatibility with existing picture scanning equipment,
- (2) Reduction in the amount of high-speed memory required, and
- (3) On-line processing of the image.

By means of the 1-D FWT the image would

be entered as a series of lines into an array of all-semiconductor processors, where processing would occur immediately. Almost at the conclusion of the scan, the FWT spectral components would be available for manipulation either for compression or image enhancement. In contrast, were the regular 2-D FWT of the image to be formed, two scans of the picture would be required -- one horizontally and one vertically. Thus for an image with 1,000 horizontal and vertical elements, one million locations of storage are implied. In turn, this suggests some type of bulk storage, immediately degrading overall speed.

Partitioning of the picture has been proposed (2, 3) as a method of overcoming the memory requirement. The image is broken up into a number of pieces of size perhaps 16×16 elements and the transform taken. An alternative described (1) is to work with narrow strips of the original picture extending from the left border to the right border. This would be consistent with conventional scanning methods and also is more suitable for some 1-D processor designs, in particular the cascade. Justification for applying the 1-D FWT follows from

Theorem: Let $f(i,j)$ represent the intensity samples of an image with $N = 2^n$ elements in each dimension and $M = N^2$ in all. Then the Walsh Transform (WT) can be calculated equivalently from

$$F(u,v) = H_N f(i,j) H_N \quad \text{or}$$

$$F(u,v)' = H_M f(i,j)',$$

where H_N and H_M are Hadamard matrices and $F(u,v)'$ and $f(i,j)'$ are column vectors formed from the rows of the corresponding matrices and with the elements ordered as shown in Fig. 7.

Discussion:

Hadamard matrices and their use in the computation of the WT are discussed in (4). The Kronecker or direct product of matrices can be illustrated as follows:

$$H_2 = \begin{bmatrix} +1 & +1 \\ -1 & -1 \end{bmatrix}$$
$$H_4 = \begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix}$$

where $H_4 = H_2 \otimes H_2$, the direct product.

$$H_4 = \begin{bmatrix} +H_2 & +H_2 \\ +H_2 & -H_2 \end{bmatrix}$$

Proceeding inductively,

$$H_{16} = \begin{bmatrix} +H_4 & +H_4 & +H_4 & +H_4 \\ +H_4 & -H_4 & +H_4 & -H_4 \\ +H_4 & +H_4 & -H_4 & -H_4 \\ +H_4 & -H_4 & -H_4 & +H_4 \end{bmatrix}$$

As an illustration of the theorem, consider the image

$$f(i,j) = \begin{bmatrix} f_{00} & f_{01} & f_{02} & f_{03} \\ f_{10} & f_{11} & f_{12} & f_{13} \\ f_{20} & f_{21} & f_{22} & f_{23} \\ f_{30} & f_{31} & f_{32} & f_{33} \end{bmatrix}$$

Then the WT can be computed either as

$$F(u,v) = H_4 F(i,j) H_4 \quad \text{or}$$

$$F(u,v) = H_{16} f(i,j).$$

Proof:

The method of proof is by explicit correspondence between terms of the two representations. Consider a single term $F(m,n)$ from the matrix $F(u,v)$. In order to evaluate it, first $f(i,j)H_4$ is formed. Its n th column can be written

$$\begin{bmatrix} +f_{00} +f_{01} -f_{02} -f_{03} \\ +f_{10} +f_{11} -f_{12} -f_{13} \\ +f_{20} +f_{21} -f_{22} -f_{23} \\ +f_{30} +f_{31} -f_{32} -f_{33} \end{bmatrix}$$

where n has been chosen equal to 2. It is important to note that the sign pattern is repetitive. If the operation is now completed by the matrix multiplication by H_4 in front, essentially the m th row of H_4 is multiplied term by term with the above column to yield ($m=1$)

$$\begin{aligned} &+(+f_{00} +f_{01} -f_{02} -f_{03}) - (f_{10} +f_{11} -f_{12} -f_{13}) \\ &+(+f_{20} +f_{21} -f_{22} -f_{23}) - (f_{30} +f_{31} -f_{32} -f_{33}). \end{aligned}$$

Thus the term $F(m,n)$ can be identified as a collection of all the terms of the matrix $f(i,j)$ grouped by rows. The sign pattern before each row is that of the m th row of H_4 and the sign pattern within each row is that of the n th row -- noting that H_4 is symmetric.

Next $H_{16} f(i,j)$ is to be evaluated, and in particular its (m,n) th term. Now if H_{16} is expressed as above as the direct product of H_4 with itself, it is immediately apparent that the sign pattern of the $4xm + n$ th row is exactly the same as that of $H_4 f(i,j)H_4$. This is sufficient to prove the equivalence. It should be remarked that the theorem is also valid for rectangular images.

Adaptive Image Partitioning

Partitioning a given image into a set of smaller squares in order to take the transform is of interest both because of hardware simplification and greater compression ratios (2,3). Although the emphasis of this paper is upon 1-D transforms, it is instructive to examine at first the use of square subregions. It has been claimed that this procedure is in a sense adaptive, meaning that "busy" regions are represented by more spectral components than rather simple regions. In what follows it will be shown that it is possible to use partitioning in a truly adaptive way in the sense that the best size of the subregions is established by the content of the picture.

An algorithm can be established so that it is not necessary to choose in advance the exact size of the subregions. The method involves building the FWT upward and outward. The starting point is

$$\begin{bmatrix} H_2 \\ H_2 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} H_2 \\ H_2 \end{bmatrix} = \begin{bmatrix} a+b+c+d & a-b+c-d \\ a+b-c-d & a-b+c+d \end{bmatrix}$$

which is the 4-point transform. Proceeding by induction, consider now the WT of a partition measuring $2N \times 2N$ is to be formed:

$$\begin{bmatrix} H_{2N} \\ H_{2N} \end{bmatrix} \begin{bmatrix} F_0 & F_1 \\ F_2 & F_3 \end{bmatrix} \begin{bmatrix} H_{2N} \\ H_{2N} \end{bmatrix}$$

But $H_{2N} = H_N \otimes H_N$ so that the above is

$$\begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix} \begin{bmatrix} F_0 & F_1 \\ F_2 & F_3 \end{bmatrix} \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix}$$

The first term of the resulting matrix is

$$H_N P_0 H_N + H_N P_1 H_N + H_N P_2 H_N + H_N P_3 H_N$$

with the other terms being similar except for the sign pattern, which is one of $+++$, $+-$, $++-$, or $+++$. Hence the algorithm can be stated that starting with the transforms of $N \times N$ regions:

$$\begin{bmatrix} H_N P_0 H_N & H_N P_1 H_N \\ H_N P_2 H_N & H_N P_3 H_N \end{bmatrix}$$

the transform of the $2N \times 2N$ region is carried out by combining corresponding terms from the four quadrants exactly as in the 4-point transform above.

This procedure is illustrated in Fig. 1. The FWT is calculated by first combining elements separated a distance 1, then a distance 2, then 4, etc. The advantage of this formulation is that the greatest correlation is expected between adjacent elements. As the process continues, more distant regions are combined. But the procedure can be halted when it no longer gives effective compression.

1	2	3	4	5	6	7	8
3	4	4	5	5	6	6	7
5	5	5	5	6	6	6	6
7	7	6	6	6	6	5	5
9	8	7	6	6	5	5	4
11	10	9	8	7	5	4	3
13	11	10	8	7	5	4	2
15	13	11	9	7	5	3	1

Fig. 1a Original Picture

10	-2	16	-2	22	-2	28	-3
-4	0	-2	0	0	0	2	0
24	0	22	0	24	0	22	0
-4	0	-2	0	0	0	2	0
38	2	30	2	22	2	16	2
-4	0	-4	0	0	0	2	0
52	4	38	4	24	4	10	4
-4	0	12	0	0	0	2	0

Fig. 1b 2x2 FWT

72	-4	-4	0	96	-4	-4	0
-12	0	-4	0	4	0	-4	0
-20	-4	-8	0	4	-4	-8	0
0	0	0	0	0	0	0	0
158	12	22	0	72	12	20	0
-14	0	-2	0	4	0	-4	0
-22	-4	-6	0	4	-4	-8	0
-2	0	2	0	0	0	0	0

Fig. 1c 4x4 FWT

398	16	34	0	62	0	2	0
-18	0	-14	0	-34	0	2	0
-34	-16	-30	0	-50	0	2	0
-2	0	2	0	-2	0	2	0
-62	-32	-50	0	-110	0	-2	0
2	0	-2	0	2	0	-2	0
2	0	-2	0	2	0	-2	0
2	0	-2	0	2	0	-2	0

Fig. 1d 8x8 FWT

Serial 1-D FWT

Based upon results of M.C. Pease, Rushforth (5) proposed a 1-D factorization of the FWT which was utilized in (1) in an implementation. It will now be shown how this factorization can be modified in analogy with the results of the preceding section to achieve building the FWT "outward" from adjacent samples. This is in contrast with the normal method of computing the FWT where the first step involves combining samples separated by a distance $N/2$. The purpose once again is to permit adaptive procedures with respect to subregion selection.

Pease expressed H_N as follows:

$$H_N = C(PCP^{-1})(P^2CP^{-2})\dots P^{n-1}CP^{-(n-1)} \\ = (CP)^n,$$

where C 's effect upon a vector is to replace successive pairs of components by the sum and difference respectively and P is the perfect shuffle. P_8 for example operating on 0,1,2,3,4,5,6,7 yields 0,4,1,5,2,6,3,7 as in shuffling cards.

It is desired to replace P by P^{-1} . This can be justified as follows:

Since C is its own inverse except for a factor of 2,

$$P^{-1}CSP = 2I.$$

Hence $(P^{-1}C)^n H_N = (P^{-1}C)^n (CP)^n = NI$.

Since H_N is its own inverse except for the factor N , it follows that

$$H_N = (P^{-1}C)^n.$$

The effect of P^{-1} on a vector can be illustrated for the case P_8^{-1} : 0,1,2,3,4,5,6,7 becomes 0,2,4,6,1,3,5,7. A second application yields 0,4,1,5,2,6,3,7. The serial implementation would be

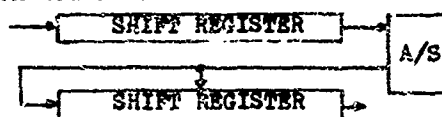


Fig. 2 Serial FWT

A/S indicates an adder/subtractor, and the Shift Registers hold the samples. The bits making up each sample can be stored in parallel SR's which are shifted simultaneously. In n steps with the data being moved back and forth the FWT is taken. Only the connections for moving from the upper SR to the lower are shown.

Application to Imagery

Use of the 1-D FWT in imagery analysis will now be briefly discussed. Assume the image measures 1024×1024 elements. It can be partitioned into strips horizontally measuring 1024×8 , say. Assuming 8-bit words and an LSI chip size capable of storing 4096 bits, 16 LSI chips are required for each SR of Fig. 2. For each section of the picture, the FWT is taken and the spectral components are manipulated either for purposes of data compression or image enhancement. For the former a threshold is established so that the weaker components are set equal to zero and the others run-length encoded. For image enhancement certain sequences are amplified and others attenuated. See Fig. 3 for 8x8 images.

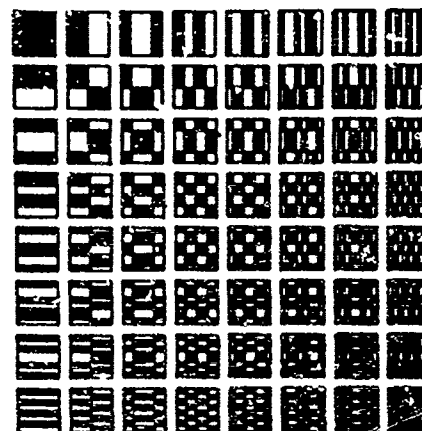


Fig. 3 Walsh Sequences

Working with the 1-D FFT these rows should be visualized placed one after another. Using the number of sign changes as equivalent to sequence, the following sequences can be assigned the patterns:

0	15	16	31	32	47	48	63
1	14	17	30	33	46	49	62
2	13	18	29	34	45	50	61
3	12	19	28	35	44	51	60
4	11	20	27	36	43	52	59
5	10	21	26	37	42	53	58
6	9	22	25	38	41	54	57
7	8	23	24	39	40	55	56

Fig. 4 Sequences

After application of the FFT the sequences occur in bit-reversed order:

0	32	16	48	15	47	31	63
4	36	20	52	11	43	27	59
2	34	18	50	13	45	29	61
6	38	22	54	9	41	25	57
1	33	17	49	14	46	30	62
5	37	21	53	10	42	26	58
3	35	19	51	12	44	28	60
7	39	23	55	8	40	24	56

Fig. 5 Bit-Reversed Order

The sequences underlined in Fig. 5 are those of the upper left quadrant of Fig. 3 and 4. These are the sequences which are usually the strongest (4). The reason for this can be seen from the algorithm demonstrated in Fig. 1. If there is redundancy in the picture, then as in Fig. 1b the very first step leads to four interleaved images which are thereafter separately processed. But of the four one tends to be far stronger when there is strong correlation from one element to the adjacent one. If every other point of Fig. 1a is chosen and the FFT formed, then after scaling the result is quite comparable to selecting every other point of Fig. 1d. Hence, in a very meaningful way, data compression by threshold sampling the FFT is related to redundancy reduction. The advantage of the adaptive procedures recommended here is that they adjust to different amounts of redundancy automatically.

One-Dimensional FFT

The extent to which the above generalizes to the FFT is of interest. However it is not true that the 1-D and 2-D FFT's are equivalent. Despite this it is still proposed to use the 1-D FFT in working with images for the reasons stated at the beginning of this report. Special purpose hardware is much more effective than general purpose computers in computing the FFT. To date these special processors have been designed to take the 1-D FFT.

Interesting enough, there is a form of the FFT in which the 1-D and 2-D FFT are equivalent. This work was carried out by Good (6,7) who pointed out the

relationship between dimensionality and the formulations of the FFT. He explored the equivalency of expressions of the form

$$A_r = \sum_r a_r w^{rs} \quad \text{and}$$

$$A_s = \sum_{r_1} \sum_{r_2} a_{r_1 r_2} w_1^{r_1 s_1} w_2^{r_2 s_2}$$

If the variables with subscript 1 range from 0 to $t_1 - 1$ and those with subscript 2 from 0 to $t_2 - 1$, where $t = t_1 t_2$, and t_1 and t_2 are relatively prime, then the expressions are equal. w , w_1 , w_2 are the corresponding principal roots of unity.

In order to prove this relationship Good made use of the Chinese remainder theorem to establish the relationship between s and the s_1 . In the two-dimensional case illustrated, r is given by

$$r = t_2 r_1 + t_1 r_2 \pmod{t}.$$

Thus if $t_1 = 3$ and $t_2 = 8$, r is shown in the following table:

	0	1	2	3	4	5	6	7
0:	0	3	6	9	12	15	18	21
1:	8	11	14	17	20	23	26	29
2:	16	19	22	25	28	31	34	37

Fig. 6 Good Relationship

Two reasons for not using this formulation can be given. The first is the order in which the samples are taken and the second is that the moduli must be relatively prime. In other words, hardware considerations suggest that it is better to use the Cooley-Tukey form despite the "twiddle" factor that makes the 1-D and 2-D versions different. The method of scanning samples then becomes

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31

Fig. 7 Proposed Scan

Having made the decision to use the 1-D FFT, the question arises as to whether it will permit the same flexibility in image manipulation as the 2-D FFT. In particular, it is worth looking at data compression. Assume that in Fig. 7 the second, third, and fourth rows are highly correlated with the first, i.e. they are almost identical. The implications can be seen from the flow graph for the FFT corresponding to Decimation in Frequency. During the first stage samples 0 and 16 will be added and subtracted, then 1 and 17, etc. This will set to zero half the spectral components. During the second stage, half the remainder will also be set to zero. Hence there results a 4:1 compression. Depending upon the correlation of samples along the first row, it is possible that additional compression can be obtained.

In a recent paper (8) the FFT was combined with partitioning of the image in order to achieve picture bandwidth compression. After some experimentation subregions of size 16×16 were explored in further depth, and actual computer simulation was conducted. For each subregion the standard deviation, L , of the picture samples was computed to within a constant and for that region the number of Fourier spectral components selected was proportional to L . In this sense the procedure was adaptive.

A problem identified with partitioning was edge effects. Because the Fourier transform is equivalent to a Fourier series expansion, differences in brightness between opposite edges act like a discontinuity requiring a substantial number of spectral components to represent it. A possible solution proposed was to add an extra row and column in order to permit interpolation, but it was noted that this would not be consistent with the FFT algorithm.

Although the use of a 1-D FFT was identified in (8) as permitting substantial hardware simplifications, it was noted that along an entire horizontal line there would surely be a larger value of L , thus requiring more spectral components to represent it, as compared with arranging the same number of samples into a more square-shaped region.

Nevertheless, the hardware considerations previously discussed make a strong case for the 1-D FFT. For example, with respect to edge effects, it would be possible to replace the right column of the picture with an interpolated column whereas this cannot be done in the center of a picture. This step in itself could possibly compensate for the disadvantage of working with a strip compared to a square.

Conclusions

Hardware simplifications can be obtained through applying 1-D rather than 2-D FFT and FFT techniques in imagery analysis. The goal is to permit on-line image manipulation by special processors. For the FFT the 1-D method and the 2-D method are shown to be equi-

valent. Algorithms for partitioning pictures adaptively for purposes of data compression and image manipulation were described and illustrated. It was possible to demonstrate the method by which threshold sampling of FFT spectral components removes redundancy from a picture.

For the FFT it is not true that the 1-D and 2-D transforms are the same when using the Cooley-Tukey algorithm. Although they are identical in the Good formulation of the FFT, it was concluded that it would be better to utilize the former. From the flow chart for Decimation in Frequency it can be shown that the 1-D FFT applied to an image and followed by threshold sampling will result in a good compression ratio. The relative efficiencies of 1-D and 2-D FFT's were briefly discussed, but it does not seem possible to resolve this question without a careful simulation study.

References

- (1) Murray, G.G., Digital Walsh Filter Design, Proceedings of the Symposium on the Applications of Walsh Functions, AD-727 000, pp. 101-105, April, 1971
- (2) Claire, E.J., Farber, S.M., Green, R.R., Practical Techniques for Transform Data Compression/Image Coding, *ibid*, pp. 2-5
- (3) Kennedy, J.D., Walsh Function Imagery Analysis, *ibid*, pp. 7-10
- (4) Pratt, W.K., Kane, J., Andrews, H.C. Hadamard Transform Image Coding, Proceedings IEEE, Vol. 57, no. 1, Jan, 1969
- (5) Rushforth, C. D., Fast Fourier-Hadamard Decoding of Orthogonal Codes, Information and Control, vol. 15, pp. 33-37, July, 1969
- (6) Good, I.J., The Interaction Algorithm and Practical Fourier Analysis, Journal of the Royal Statistical Soc., Vol. 20, no. 2, pp. 361-372, 1958
- (7) ———, The Relationship Between Two Fast Fourier Transforms, IEEE Trans. on Computers, Vol. C-20, no. 3, pp. 310-317, March, 1971
- (8) Anderson, G. B., Huang, T.S., Piecewise Fourier Transformation for Picture Bandwidth Compression, IEEE Trans. Communication Technology, Vol. Com-19, no. 2, pp. 133-140, April, 1971

QUANTIZATION NOISE CONSIDERATIONS IN WALSH TRANSFORM IMAGE PROCESSING

Guner S. Robinson
Member of Technical Staff
COMSAT Laboratories
Clarksburg, Md. 20734, USA

Introduction

The use of conventional pulse-code modulation techniques to transmit television signals requires a very wideband transmission medium. Therefore, new bandwidth compression (or, equivalently, bit-rate reduction) techniques have been explored for digital transmission of television signals.

Reduction of the bit rate required for the transmission of a television channel is possible if the coefficients obtained from a suitable orthogonal transform of the television signal are properly coded. The advantage of using an orthogonal transformation depends on the suitability of the transformation for the representation of the signal and the simplicity of the implementation. The overall performance can then be judged by using conventional PCM as a reference.

With the development of high-speed digital computers, various transformations have recently been explored by computer simulation for possible bit-rate reduction. The well-known Fourier transform, which can be computed very rapidly, is typical of the orthogonal transforms which are being studied. Although the FFT algorithm is very efficient, its implementation is not simple. However, the Fourier transform of an image can be practically determined by using a coherent optical system.

Another transformation which can be computed very rapidly is the discrete Walsh or Hadamard transformation which does not require any multiplication operations. Alternatively, the television signal may be considered as a random process and the eigenvectors derived from the covariance matrix of the process may be used to represent the signal. This transform is known as the Karhunen-Loève transform.

The application of linear transformation and block quantization to image bandwidth reduction problems has been studied by several researchers. A partial bibliography pertaining to Walsh-Hadamard processing is given at the end of this paper.

In this paper, the application of Walsh transform processing has been investigated by using a flying spot scanner and a general-purpose computer. A full image frame of 512×512 elements has been subdivided into 1024×16 blocks, which are processed by the two-dimensional Walsh transformation. To minimize the mean-square error caused by quantization, the bit distribution in the transform domain is made in accordance with the variances of the coefficients, with the constraint that the total number of bits should not exceed a given maximum. This maximum is determined by considering PCM transmission of the same image. The error caused by intensity quantization has been computed by comparing the input and output intensity values for each line of the entire frame. For the Walsh transform the error caused by quantization of the coefficients has been computed by comparing the input and output intensity values over each block of the entire frame. A reduction in the quantizing noise has been verified by using transform domain processing instead of spatial domain processing.

Applications of the two-dimensional discrete Fourier and Karhunen-Loève transformations have also been investigated by using the same block size of $N = 16$ samples. The Karhunen-Loève transformation is most effective in reducing the contours resulting from quantizing noise, but its implementation is most complex. The Walsh-Hadamard transformation is the simplest to implement and significantly reduces the contours at low bit rates.

Two-Dimensional Discrete Walsh Transformation

Let $f(x,y)$ represent a two-dimensional square array of values obtained by sampling the brightness of an image at $N \times N$ points. (The value used for N is 16.) Assume that $F(j,k)$ is the two-dimensional Walsh-Hadamard transform of $f(x,y)$. Mathematically, such a two-dimensional discrete Walsh transform pair is defined as

$$F(j,k) = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \text{ wal}(j,x) \text{ wal}(k,y)$$

$$j, k = 0, 1, \dots, N-1 \quad (1)$$

$$f(x,y) = \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} F(j,k) \text{wal}(j,x) \text{wal}(k,y)$$

$$x,y = 0, 1, \dots, N-1 \quad (2)$$

where $\text{wal}(j,x)$ and $\text{wal}(k,y)$ represent the one-dimensional Walsh functions in the x and y directions, and j and k are the numbers of zero crossings of the Walsh functions in the x and y directions, respectively. The coefficient corresponding to $j = k = 0$,

$$F(0,0) = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)$$

yields the average brightness of the $N \times N$ block and is identical to the zero spatial frequency term in the two-dimensional discrete Fourier domain.

The application of the two-dimensional Walsh transform can be visualized as a comparison of the brightness pattern of each 16×16 image block with the various two-dimensional Walsh basis patterns for $N = 16$. Walsh basis patterns for $N = 8$ are shown in Figure 1. The coefficient corresponding to the basis $\text{wal}(1,x) \text{wal}(0,y)$ represents a comparison of the brightness of the left and right halves of the $N \times N$ block. The coefficient corresponding to the basis $\text{wal}(0,x) \text{wal}(1,y)$ represents a comparison of the brightness of the lower and upper halves of the image block. Finally, the coefficient corresponding to the $\text{wal}(N-1,x) \text{wal}(N-1,y)$ basis represents a comparison of the image block with an $N \times N$ checkerboard design, as shown in the upper right corner of Figure 1. This coefficient,

$$F\left(\frac{N}{2}, \frac{N}{2}\right) = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} (-1)^{x+y} f(x,y)$$

is identical to the spatial folding frequency term in two-dimensional discrete Fourier transform processing.

Since the transformation kernel,

$$\text{wal}(j,x) \text{wal}(k,y)$$

is separable, the transform can be applied as two consecutive one-dimensional transforms. Therefore, transformation coefficients of each block are computed by applying the one-dimensional Walsh transform, first in the x direction to every line, and then in the y direction to every column of the first transformation. If the Hadamard transform in natural order is used in the x direction, the coefficients should be sequency-ordered before it is applied in the y direction. The resulting $N \times N$ coefficients, which represent an $N \times N$ block

of image in the Walsh domain, are quantized by using a certain bit distribution, and the processed image is reconstructed as two consecutive one-dimensional transforms, first in the k direction and then in the j direction.

Quantization of the Walsh Coefficients

Thresholding magnitudes of the coefficients in the transform domain makes bit-rate reduction possible. That is, the coefficient is transmitted if its energy is above the threshold and not transmitted if its energy is below the threshold.

For high-quality image transmission every coefficient must be considered in terms of its importance in image reconstruction. Since the amount of information contained in each coefficient is proportional to its energy, bit distribution and hence allotment of the quantization levels should be made in accordance with the variances of the coefficients. This procedure also minimizes the rms error at any given bit rate.¹

The total quantizing noise caused by quantized transmission of the $N \times N$ block of image is

$$\overline{\epsilon^2} = \frac{1}{N^2} \sum_{i=0}^{N^2-1} K_i \frac{\sigma_i^2}{2^{2n_i}} \quad (3)$$

where K_i is a constant depending on the quantizer structure used for quantizing the i th coefficient, σ_i^2 is the variance of the i th coefficient, and n_i is the number of bits assigned to the i th coefficient.

Bits are distributed in the transform domain so that the total quantizing noise given by equation (3) is minimized. The average number of bits,

$$\overline{n_i} = \frac{1}{N^2} \sum_{i=0}^{N^2-1} n_i \quad (4)$$

is determined by considering PCM transmission of the $N \times N$ block of image.

In PCM transmission of an image, the spatial domain samples are quantized by using M bits per sample. The total number of bits required to transmit an $N \times N$ block of frame is then

$$B = N^2 M \quad (5)$$

Typical values chosen for M are seven, four, and two bits per sample; thus the total numbers of bits used for $N = 16$ are $B = 1792, 1024$, and 512 , respectively. In order to evaluate the

signal-to-quantization noise performance of orthogonal transform processing on the same image, the coefficients are assigned bits according to their variances and the constraint that the total numbers of bits are 1792, 1024, and 512, resulting in averages of seven, four, and two bits per picture element, respectively.

Experimental Results

A moonscape slide has been scanned by a flying spot scanner to produce an array of $L \times L$ ($L = 512$) uniformly spaced samples of image brightness. Next, a general-purpose computer has been used to perform spatial domain or transform domain processing on this array of data. Then, the processed image has been reconstructed by the flying spot scanner.

The moonscape image has been processed by PCM at three different bit rates: $M = 7, 4$, and 2 bits per sample. Figures 2a, 2b, and 2c are the resulting pictures for each of these cases. Uniform quantizing with a mid-riser has been employed in each case, since this prevents the loss of one quantizing step. The quantizing step size has been referenced to the full black-to-white range observed at the flying spot scanner output.

Walsh transformation processing has been accomplished on small $N \times N$ ($N = 16$) blocks. Thus there are $p = (512 \times 512) / (16 \times 16) = 1024$ such blocks. The Walsh transform of each block has been computed to obtain the coefficients which represent the image in the transform domain. At the receiving side, the spatial domain samples are reconstructed from the quantized coefficients by using an inverse transformation.

Bit distribution in the Walsh domain is determined by coefficient variances which are computed by ensemble averaging over the total number of blocks. Three different values are assigned to n_i : 7, 4, and 2. The bit distributions for these three values are given in Tables 1, 2, and 3, and the resulting images are shown in Figures 3a, 3b, and 3c. The zero spatial frequency (or frequency) term, $F(0,0)$, has the highest variances. Therefore, it is assigned the highest number of quantization levels in order to prevent the appearance of block structure in the reconstructed image. The number of bits assigned to $F(0,0)$ appears in the upper left corner of each table. For the three cases considered, the numbers of bits assigned to $F(0,0)$ are 14, 11, and 9 bits, respectively.

In PCM processing, the error caused by intensity quantization is computed by comparing the input and output intensity values for each line of the entire frame. For the discrete Walsh transform, the

error caused by quantization of the coefficients is computed by comparing the input and output intensity values over each block of the entire frame. PCM is used as a basis of comparison to compute the resulting reduction in the quantizing noise ratio, S/N_Q . For the three cases considered, the S/N_Q was reduced by 7.9, 3.1 and 1.6 dB, respectively. It was observed that the choice of the quantizer parameters is critical in obtaining a high improvement in S/N_Q .

In order to obtain the maximum signal-to-quantizing noise ratio, the amplitude loading factor of the uniform quantizer in the transform domain should be dependent on the number of bits assigned to the coefficients. If σ_i^2 is the variance of the i th coefficient, the quantizer spread (defined as the distance between minimum and maximum levels of the quantizer) is given by

$$Q_S = 2\alpha_i \sigma_i \quad (6)$$

where α_i is the amplitude loading factor of the i th quantizer. Table 4 gives the S/N_Q improvement when a constant amplitude loading factor is used for every coefficient.

An alternative method of processing at an average rate of four bits per sample was also investigated. This method, which consisted of assigning new optimum values of α_i (given in the appendix) for each of the bit assignments, resulted in an S/N_Q improvement of 4.1 dB, which is 1.0 dB higher than the improvement obtained for $\alpha = 4$.

Summary and Conclusions

Image processing using two-dimensional discrete Walsh, Fourier, and Karhunen-Loève transformations is performed on a moonscape slide at bit rates averaging seven, four, and two bits per picture element. At all bit rates, a reduction in the quantizing noise is verified by using transform domain processing instead of spatial domain processing. A very important property of transform processing that is not revealed in S/N_Q improvement is the subjective effect of contouring. It is basically true of all of the orthogonal transformation processing methods that at low bit rates the effect of contouring is reduced because each picture element in the transformed image is a weighted average of N two-dimensional orthogonal functions.

Appendix

The range of a quantizer (referred to as the quantizer spread) has been given by equation (6). If the i th Walsh transform coefficient is assigned n_i bits, then the step size

$$E_0 = \frac{Q_s}{2^{n_i}}$$

of a uniform quantizer is

$$E_0 = \frac{2\alpha_i \sigma_i}{2^{n_i}} \quad (A-1)$$

If there is a large number of steps, then the mean-square quantizing noise error is one-twelfth of the square of the step size.² That is,

$$\frac{E_0^2}{\epsilon^2} = \frac{E_0^2}{12} = \frac{1}{3} \frac{\alpha_i^2 \sigma_i^2}{2^{2n_i}} \quad (A-2)$$

When there is a finite number of steps, 2^{n_i} , the expression for the mean-square quantizing noise error is more involved, since the probability distribution of the coefficients should be considered. The probability distribution of the coefficients can be assumed to be Gaussian, since the coefficients are obtained from a linear combination of spatial domain samples.³

$$p(x) = \frac{1}{\sqrt{2\pi} \sigma_i} \exp(-x^2/2\sigma_i^2) \quad (A-3)$$

The exact analytical expression for quantizing noise power is known for stationary Gaussian signals.⁴ The quantizing noise power, N_Q , which is composed of two parts, granular noise power, N_G , and overload noise power, N_O , can be computed as follows:

$$\frac{N_G}{\sigma_i^2} = \frac{2}{3} \frac{\alpha_i^2}{2^{2n_i}} \operatorname{erf}(\alpha_i) \quad (A-4)$$

$$\frac{N_O}{\alpha_i^2} = (1 + \alpha_i^2) [1 - 2 \operatorname{erf}(\alpha_i)] - \sqrt{\frac{2}{\pi}} \alpha_i e^{-\alpha_i^2/2} \quad (A-5)$$

$$N_Q = N_G + N_O \quad (A-6)$$

where

$$\operatorname{erf}(\alpha_i) = \frac{1}{\sqrt{2\pi}} \int_0^{\alpha_i} e^{-x^2/2} dx$$

For large α_i , equation (A-4) yields equation (A-2). This term is what is generally known as quantizing noise because, when a suitable value of α_i is used, the overload noise is less than the granular noise and is therefore ignored.

Equations A-4, A-5, and A-6 are used to compute N_Q/σ_i^2 and the signal-to-quantizing noise ratio

$$(S/N_Q)_i = 10 \log_{10} \left(\frac{\sigma_i^2}{N_Q} \right)$$

for various n_i . Figure A-1 shows the $(S/N_Q)_i$ values versus α_i for various n_i . The optimum α_i , which maximizes the signal-to-quantizing noise ratio determined from these curves, is given in Table A-1.

Acknowledgment

This paper is based partly upon work performed at COMSAT Laboratories under Corporate sponsorship and partly upon work performed under the sponsorship of NASA. The author would like to thank Dr. S. J. Campanella who was the project manager and R. L. Granger who performed the computer programming.

References

1. Campanella, S. J. and Robinson, G. S., "A Comparison of Orthogonal Transformations for Digital Speech Processing," IEEE Transactions on Communications Technology, Vol. COM-19, December 1971, pp. 1045-1050.
2. Bennett, W. R., "Spectra of Quantized Signals," Bell System Technical Journal, Vol. 27, July 1968, pp. 446-472.
3. Pratt, W. K., Kane, J. & Andrews, H. C., "Hadamard Transform Image Coding," Proceedings of the IEEE, Vol. 57, January 1969, pp. 58-68.
4. Abate, J. E., "Linear and Adaptive Delta Modulation," Proceedings of the IEEE, Vol. 55, No. 3, March 1967, pp. 298-308.

Bibliography on Image Processing by Orthogonal Transformations

1. Agarwal, V. K. & Stephens, T. J., Jr., "On-Board Data Processor-Picture Bandwidth Compression," IRAD Report, No. R-7333.3-32, TRW System Group, California, February 1970.
2. Alexandridis, N. A. and Klinger, A., "Real-Time Walsh-Hadamard Transformation," IEEE Transactions on Computers (Short Notes) Vol. C-21, No. 3, March 1972, pp. 288-292.
3. Anderson, G. B. & Huang, T. S., "Piecewise Fourier Transformation for Picture Bandwidth Compression," IEEE Transactions on Communications Technology, Vol. COM-19, No. 2, April 1971.

4. Andrews, H. C., Computer Techniques in Image Processing, Academic Press, New York: 1970.
5. Andrews, H. C., "Fourier and Hadamard Image Transform Channel Error Tolerance," Proceedings of the UMR-Mervin J. Kelly Communications Conference, University of Missouri Rolla, Missouri, October 1970, (paper 10-4).
6. Andrews, H. C. & Pratt, W. K., "Digital Image Transform Processing," Proceedings of the 1970 Symposium on the Applications of Walsh Functions, pp. 183-194, Washington, D. C. April 1970.
7. Andrews, H. C. & Pratt, W. K., "Television Bandwidth Reduction by Encoding Spatial Frequencies," Journal Society of Motion Picture and Television Engineers, Vol. 77, December 1968, pp. 1279-1281.
8. Bowyer, D., "Walsh Functions, Hadamard Matrices and Data Compression," Proceedings of the 1971 Symposium on the Applications of Walsh Functions, Washington, D. C., April 1971.
9. Claire, E. J., Farber, S. M. & Green, R. R., "Practical Techniques for Transform Data Compression," Proceedings of the 1971 Symposium on the Applications of Walsh Functions, Washington, D. C., April 1971.
10. Enomoto, H. & Shibata, K., "Orthogonal Transform Coding System for Television Signals," Proceedings of the 1971 Symposium on the Applications of Walsh Functions, Washington, D. C., April 1971.
11. Fino, B. J., "Traitement d'Images par Transformations Orthogonales," CNET Note Technique, ITD/TTI/52, September 1971.
12. Habibi, A. & Wintz, P. A., "Image Coding by Linear Transformation and Block Quantization," IEEE Transactions on Communications Technology, Vol. COM-19, No. 1, pp. 50-62, February 1971.
13. Kennedy, J. D., "Walsh Function Imagery Analysis," Proceedings of the 1971 Symposium on the Applications of Walsh Functions, Washington, D. C., April 1971.
14. Kennedy, J. D., Clark, S. J. & Parkyn, W. A., "Digital Imagery Data Compressor Techniques," McDonnell Douglas Astronautics Company, Report MDC G0402, California, January 1970.
15. Lanquau, H. J. & Slepian, D., "Some Computer Experiments in Picture Processing for Bandwidth Reduction," Bell System Technical Journal, May-June 1971, pp. 1525-1540.
16. Parkyn, Jr., W. A., "Digital Image Processing Aspects of the Walsh Transform," Proceedings of the 1970 Symposium on Applications of Walsh Functions, Naval Research Laboratory, Washington, D. C., April 1970, pp. 152-156.
17. Ponchin, J., "Utilisation de la Transformation de Hadamard pour le Codage et la Compression de Signaux d'Image," Annales des Telecommunications, July-August 1971, pp. 235-251.
18. Pratt, W. K., "Spatial Domain Coding of Color Images," IEEE Transactions on Communication Technology, Vol. COM-19, December 1971, pp. 980-992.
19. Pratt, W. K. & Andrews, H. C., "Transform Image Coding," USCEE Report No. 387, University of California, March 1970.
20. Pratt, W. K., Kane, J. & Andrews, H. C., "Hadamard Transform Image Coding," Proceedings of the IEEE, Vol. 57, January 1969, pp. 58-68.
21. Shibata, K., Ohira, T. & Terauchi, S., "Color Television Signal Orthogonal Transform PCM Terminal Experimental Equipment," IEECJ Transactions on Communications Systems, (Japan), CS70-47, July 29, 1970.
22. Wishner, H. D., "Designing Special-Purpose Digital Image Processor," Computer Design, Vol. 11, No. 2, February 1972, pp. 71-76.
23. Woods, J. W., "Video Bandwidth Compression by Linear Transformation," Quarterly Progress Report No. 91, M.I.T. Research Laboratory of Electronics, October 15, 1968, pp. 219-224.

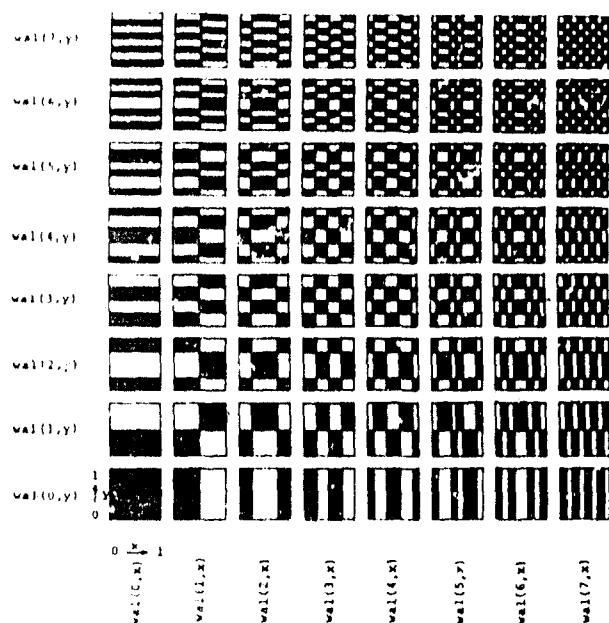


Figure 1. Two Dimensional Walsh-Hadamard Basis for $N=8$.

Black areas represent $+1/N^2$
and white areas represent $-1/N^2$.

Table 2. Optimum Bit Distribution in Two-Dimensional Walsh-Hadamard Domain (average four bits per sample)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	11	8	7	7	6	6	6	6	4	4	4	4	5	5	5	5
1	9	8	7	7	6	6	6	5	4	4	4	4	4	5	5	4
2	8	7	7	6	5	6	6	5	3	4	4	4	4	4	4	4
3	8	7	6	6	5	5	5	5	3	4	4	4	4	4	4	3
4	7	6	6	6	5	5	5	4	3	3	4	4	4	4	4	3
5	7	6	6	6	5	5	5	4	3	3	3	4	4	4	4	3
6	7	6	6	5	5	5	5	4	3	3	3	3	3	3	4	3
7	6	6	5	5	4	4	4	4	3	3	3	3	3	3	3	2
8	5	5	5	4	4	4	4	3	2	3	3	3	3	3	3	2
9	5	5	5	4	4	4	4	3	2	3	3	3	3	3	3	2
10	5	5	5	4	4	4	4	3	2	2	3	3	3	3	3	2
11	6	5	5	4	4	4	4	3	2	2	3	3	3	3	2	2
12	6	5	5	4	4	4	4	3	2	2	2	3	3	3	3	2
13	6	5	5	4	4	4	4	3	2	2	2	2	2	3	2	2
14	6	5	4	4	3	3	3	3	2	2	2	2	2	2	2	2
15	5	4	4	3	3	3	3	2	1	1	1	2	2	1	1	1

Table 1. Optimum Bit Distribution in Two-Dimensional Walsh-Hadamard Domain (average seven bits per sample).

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	11	10	10	9	9	9	9	7	7	7	7	8	8	8	8
1	12	11	10	10	9	9	9	8	7	7	7	7	7	8	8	7
2	11	10	10	9	8	9	9	8	6	7	7	7	7	7	7	7
3	11	10	9	9	8	8	8	8	6	7	7	7	7	7	7	6
4	10	9	9	9	8	8	8	7	6	6	7	7	7	7	7	6
5	10	9	9	9	8	8	8	7	6	6	6	7	7	7	7	6
6	10	9	9	8	8	8	8	7	6	6	6	6	6	6	7	6
7	9	9	8	8	7	7	7	7	6	6	6	6	6	6	6	5
8	8	8	8	7	7	7	7	6	5	6	6	6	6	6	6	5
9	8	8	8	7	7	7	7	6	5	6	6	6	6	6	6	5
10	8	8	8	7	7	7	7	6	5	5	6	6	6	6	6	5
11	9	8	8	7	7	7	7	6	5	5	6	6	6	6	5	5
12	9	8	8	7	7	7	7	6	5	5	6	6	6	6	6	5
13	9	8	8	7	7	7	7	6	5	5	5	5	5	6	5	5
14	9	8	7	7	6	6	6	6	5	5	5	5	5	5	5	5
15	8	7	7	6	6	6	6	5	4	4	4	4	4	4	4	4

Table 3. Optimum Bit Distribution in Two-Dimensional Walsh-Hadamard Domain. (average two bits per sample)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	9	6	5	5	4	4	4	4	2	2	2	2	3	3	3	3
1	7	6	5	5	4	4	4	3	2	2	2	2	2	2	3	2
2	6	5	5	4	3	4	3	3	1	2	2	2	2	2	2	2
3	6	5	4	4	3	3	3	3	1	2	2	2	2	2	2	1
4	5	4	4	4	3	3	3	2	1	1	2	2	2	2	2	1
5	5	4	4	4	3	3	3	2	1	1	1	2	2	2	2	1
6	5	4	4	3	3	3	3	2	1	1	1	1	1	1	1	1
7	4	4	3	3	2	2	2	2	1	1	1	1	1	1	1	0
8	3	3	3	2	2	2	2	2	1	0	1	1	1	1	1	0
9	3	3	3	2	2	2	2	2	1	0	1	1	1	1	1	0
10	3	3	3	2	2	2	2	2	1	0	0	1	1	1	1	0
11	4	3	3	2	2	2	2	2	1	0	0	1	1	1	1	0
12	4	3	3	2	2	2	2	2	1	0	0	0	1	1	1	0
13	4	3	3	2	2	2	2	2	1	0	0	0	0	0	0	0
14	4	3	2	2	1	1	1	1	0	0	0	0	0	0	0	0
15	3	2	2	1	1	1	0	0	0	0	0	0	0	0	0	0



a. Hadamard, Average 7 bits/sample, $\alpha=8$



b. Hadamard, Average 4 bits/sample, $\alpha=4$



c. Hadamard, Average 2 bits/sample, $\alpha=4$



a. PCM, M=7 bits/sample



b. PCM, M=4 bits/sample



c. PCM, M=2 bits/sample

Figure 2. Moonscape Processed by PCM.

Figure 3. Moonscape Processed by Walsh-Hadamard Transform.

Table 4. Signal-to-Quantizing Noise Ratio Improvement of Two-Dimensional Orthogonal Transforms over PCM as Applied to Moonscape Slide.

Average Number of Bits used in the Transform Domain \bar{n}_1 (bits/sample)	Amplitude Loading Factor of the Quantizer in the Transform Domain α_1	Improvement in S/N_Q with respect to PCM (dB)		
		Fourier	Hadamard	Karhunen- Loève
2	4	9.8	7.9	13.7
4	4	4.0	3.1	4.8
7	8	3.9	1.6	9.9

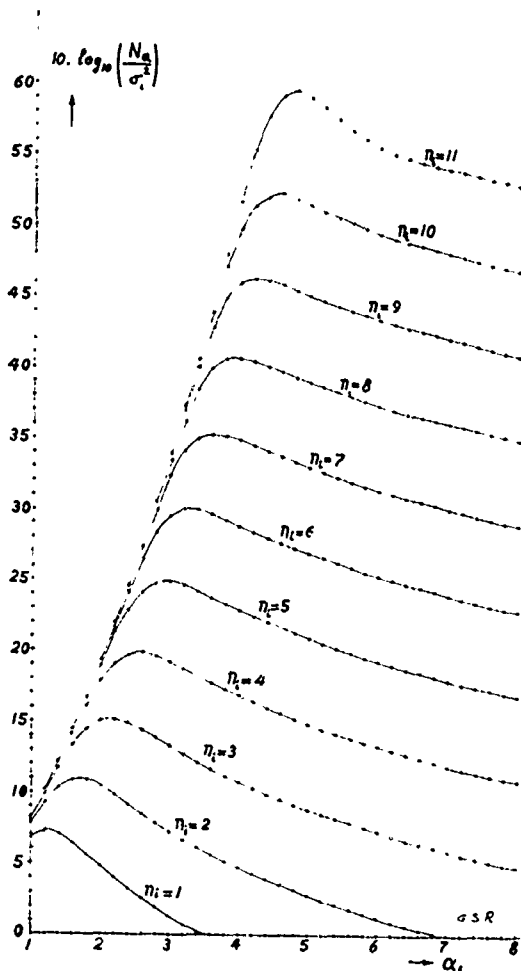


Figure A-1. Signal-to-Quantizing Noise vs Amplitude Loading Factor, α , for Various Bit Assignments

Table A-1. Optimum Amplitude Loading Factor of the Quantizer in the Walsh-Hadamard Transform Domain

Number of Bits Assigned to the i-th Coefficient n_i (bits/sample)	Optimum Amplitude Loading Factor for the Uniform Quantization of i-th Coefficient	
	α_i optimum	α_i used
1	1.25	1.5
2	1.8	2.0
3	2.1	2.5
4	2.5	3.0
5	3.0	3.4
6	3.3	3.8
7	3.7	4.1
8	4.0	4.4
9	4.3	4.7
10	4.5	4.9
11	4.7	5.1

SPATIAL SUBSAMPLING FOR THE TRANSFORM CODING OF IMAGES*

Clifford Reader
University of Southern California
Department of Electrical Engineering
Los Angeles, California

Introduction

The implementation of orthogonal transformation over images composed of a large array of pixels is an arduous task. Earlier work [1] has shown that, if a degree of error may be tolerated then the image may be divided into blocks each of which may be separately transformed. Typically for an image of 256 x 256 pixels, blocks may be selected of size 16 x 16 pixels. The problem which arises is that for high orders of redundancy reduction, the edges of the blocks and the low frequency reconstruction patterns become visible. The object of this study was to minimize this spurious block pattern and checkering effect.

Subsampling Patterns

The desired effect of blurring the edges of the blocks may be attained by spatially sub-sampling the image prior to dividing it into the blocks. Preferably (from the standpoint of blurring the edges), this should be done in as random a way as possible. However, the redundancy reduction properties of the transform derive from exploitation of the inter-element correlations and any sub-sampling scheme is bound to reduce the correlation between samples in the rearranged array. Thus, there will be a trade-off between the reduction of image quality due to this effect and any improvement due to the elimination of discrete errors. The pattern chosen for evaluation was formed from a Latin square.

A Latin square is an $n \times n$ array of the integers 1 to n such that each integer appears once and only once in each row and column of the array. The simplest non-trivial example is the 2 x 2 Latin square:

$$\begin{array}{cc} 1 & 2 \\ 2 & 1 \end{array}$$

Repeated over a large array this is equivalent to the checkerboard pattern. Thus the image could be separated into two images according to the pattern and the 16 x 16 blocks selected from the resultant array. Two examples are shown out of the 24 possible 4 x 4 Latin

squares of interest.

$$\begin{array}{cc} \text{(i)} & \begin{array}{cccc} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \\ 3 & 4 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{array} & \text{(ii)} & \begin{array}{cccc} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \\ 2 & 1 & 4 & 3 \\ 3 & 4 & 1 & 2 \end{array} \end{array}$$

Application of these will of course lead to four images, but the inter-element correlation of each of those images will be much reduced. The basic scheme may be modified by selecting the elements in sub-blocks of $m \times m$ elements ($m = 1, 2, 4, 8$). This has the advantage that the inter-element correlation will be high across the sub-block but the edges of these sub-blocks will appear just as those of the larger blocks. An investigation was made to determine the statistics of the Latin squares and hence choose those most likely to be of use.

Latin Square Correlation and covariance matrices

The assumption was made that the image may be modelled as a first order Markov process with inter-element correlation:

$$C_{ij} = \rho^r \quad i \neq j \quad r \text{ integer.} \\ = 1 \quad i = j$$

For sub-sampling according to a 4 x 4 Latin square, the row correlation matrix becomes:

$$\begin{bmatrix} 1 & \rho^4 & \rho^8 & \rho^{12} & \dots & \rho^{60} \\ \rho^4 & 1 & \rho^4 & \rho^8 & & \vdots \\ \rho^8 & \rho^4 & 1 & \rho^4 & & \\ \vdots & & & & & \\ \rho^{60} & \dots & & & & 1 \end{bmatrix}$$

To model the column correlation matrix a further assumption must be made - that the correlation between elements of a different row and column is the multiple of the correlations along the rows and down the columns between the samples. For sub-sampling according to the Latin square:

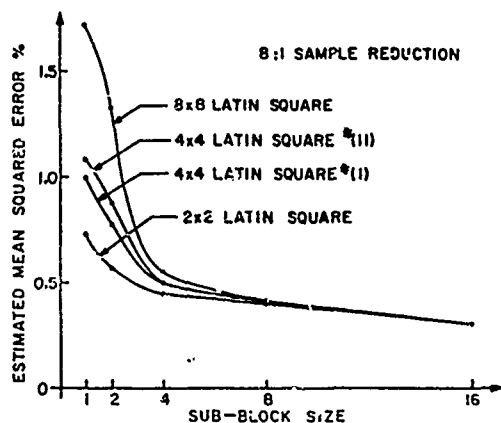
*This research was supported by the Advanced Research Projects Agency of the Defense and was monitored by the Air Force Eastern Test Range under Contract No. FO 8606-72-C-0008.

1 2 3 4
2 1 4 3
3 4 2 1
4 3 1 2

the column correlation matrices of the samples selected in the pattern of the "ones" is then:

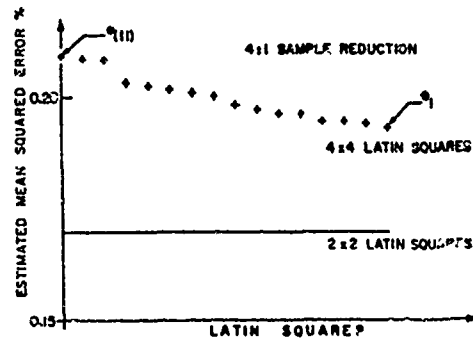
$$\begin{bmatrix} 1 & \rho^2 & \rho^5 & \rho^5 & \rho^4 & \rho^6 & \rho^9 & \dots \\ \rho^2 & 1 & \rho^3 & \rho^3 & \rho^4 & \rho^4 & \dots \\ \rho^5 & \rho^3 & 1 & \rho^2 & \rho^5 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

Software was written to calculate these matrices, their transforms and hence the variances of the transformed reordered samples. Estimations of mean squared error made for various zonal filtering operations on the Hadamard transform domain are summarized in graph 1.



Graph 1. Estimated mean squared error versus sub-block size.

As would be expected there is little difference between the different Latin squares except for small sub-block sizes where the average inter-element correlation begins to decrease. It was found that a sub-block size of 4 x 4 will give a moderately worsened mean square error, while for the 4 x 4 Latin square offering a suitably random pattern. Graph II shows the variation of estimated mean squared error against the Latin square patterns for a sub-block size 4 x 4 and a 4:1 sample reduction.



Graph II. Estimated mean squared error for various Latin squares.

The Latin square number (i) is the one with least mean squared error, number (ii) the one with greatest. This is apparent if diagonally adjacent elements are marked to indicate high correlation:

(i)	1	2	3	4	(ii)	1	2	3	4
	X		X				X		
	2	1	4	3		4	3	2	1
			X						
	3	4	1	2		2	1	4	3
	X		X				X		
	4	3	2	1		3	4	1	2

Even so, the difference in expected mean squared error appears small enough to leave freedom to choose the Latin square most suited to reordering the data.

Experimental Results

Results were obtained for sub-sampling according to the checkerboard pattern and the two Latin square patterns (i) & (ii) for various sub-block sizes. The transform domains were quantized by an optimised scheme to give a bandwidth reduction of 8:1 for the Hadamard transforms; 12:1 for the Karhunen - Loeve transforms. The quantization scheme used allocated bits to samples in the transform domain according to the variances of those samples. Two bit allocation schemes are shown in Figure 1 which illustrate the effect of spatial sub-sampling

on the statistics of the transform domain.

Figure 1. Bit allocation schemes for transform domain quantization.

2	4	8	7	5	4	4	4	3	3	3	3	3	3
2	7	5	3	3	2	2	2	2	2	0	0	0	0
3	4	3	2	2	0	0	0	0	0	0	0	0	0
3	3	2	2	0	0	0	0	0	0	0	0	0	0
5	3	2	0	0	0	0	0	0	0	0	0	0	0
4	2	0	0	0	0	0	0	0	0	0	0	0	0
4	2	0	0	0	0	0	0	0	0	0	0	0	0
3	2	0	0	0	0	0	0	0	0	0	0	0	0
3	2	0	0	0	0	0	0	0	0	0	0	0	0
3	2	0	0	0	0	0	0	0	0	0	0	0	0
3	2	0	0	0	0	0	0	0	0	0	0	0	0
3	2	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0

(a) Hadamard domain, no sub-sampling.

2	4	7	4	3	3	3	2	2	2	2	2	2	2
2	4	5	4	2	2	2	2	0	0	0	0	0	0
2	4	4	5	2	2	2	2	0	0	0	0	0	0
2	4	4	4	2	2	2	2	0	0	0	0	0	0
4	2	0	0	0	0	0	0	0	0	0	0	0	0
3	2	0	0	0	0	0	0	0	0	0	0	0	0
3	2	0	0	0	0	0	0	0	0	0	0	0	0
3	2	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0

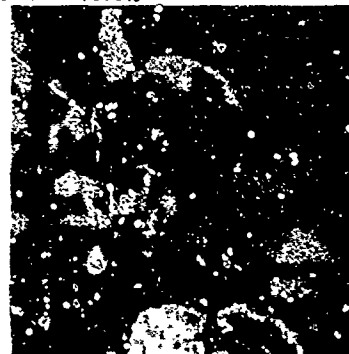
(b) Hadamard domain, sub-sampling with Latin square (i); sub-block size 4 x 4.

Sub-sampling inherently reduces sample correlations and leads to a lesser degree of energy compaction around the origin of the transform domain.

Figure 2. Images processed without sub-sampling.



(a) Hadamard transform; mean squared error = 0.161%



(b) Karhunen-Loeve transform; mean squared error = 0.186%

The checking effect is pronounced and the block edges are visible over part of the image which was transformed with the Karhunen-Loeve transform. Figure 3 shows the results of checkerboard sub-sampling with the Hadamard transform.

Figure 3. Effect of checkerboard sub-sampling.



(a) Hadamard transform, sub-block size 1 x 1; mean squared error = 0.28%



(b) Hadamard transform, sub-block size 8 x 8; mean squared error = 0.186%

If the sub-block size is 1×1 - a single pixel - then the image is degraded. This is due to the severe reduction of inter-element correlation which the quantization scheme cannot accommodate. However, for an 8×8 sub-block size the discrete checking effect, apparent in the image which was not sub-sampled, has been reduced while overall image clarity has diminished a little. Figure 3 (c) shows the effect of checkerboard sub-sampling with the Karhunen-Loeve transform.



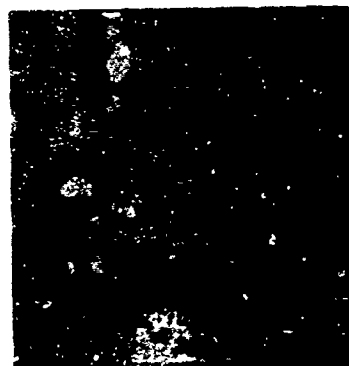
(c) Karhunen-Loeve transform, sub-block size 2×2 ; mean squared error = 0.312%.

It was found that use of 8×8 sub-blocks resulted in the visibility of those blocks and for a 2×2 sub-block size the image is noticeably degraded. This is a reflection of the highly optimized nature of the transform; any reduction of correlation in the original image results in a poorer quality output image. The effect of sub-sampling with Latin squares is shown in Figure 4.

Figure 4. Effect of 4×4 Latin square sub-sampling with the Hadamard transform.



(a) Latin square (i), sub-block size 8×8 ; mean squared error = 0.189%



(b) Latin square (ii), sub-block size 8×8 ; mean squared error = 0.182%.



(c) Latin square (ii), sub-block size 4×4 ; mean squared error = 0.208%

Similar to the results for checkerboard sub-sampling, these results indicate the variation in the size of the checkering pattern for different sub-sampling patterns.

Conclusion

There is a modest improvement in subjective image quality when spatial sub-sampling is used in conjunction with the Hadamard transform owing to minimization of the checkering effect. The visibility of that effect may be controlled by the choice of sub-sampling pattern. Application of sub-sampling schemes to the Karhunen-Loeve transform coding of images leads only to a reduction of image quality. Use of a large sub-block size results in visibility of those sub-blocks and use of a small sub-block size reduces the effective inter-element correlation sufficiently to impair the efficiency of the transform. It is expected that there would be an improvement in image quality for the sub-optimum Fourier and Slant transforms.

Reference

- [1] W. K. Pratt and H. C. Andrews,
"Transform Coding of Images,"
USCEE Report 387, Pages 56-57
(March, 1970)

GENERALIZED SAMPLING INTERPRETATION OF HADAMARD AND HAAR TRANSFORMS

Heinz H. Schreier
Grumman Aerospace Corporation
Bethpage, New York 11714

Introduction

The use of Hadamard and Haar transformations in signal processing applications such as image coding, [1] voice transmission, [2, 3] and data reduction, [4, 5] is discussed and analyzed in extant literature. Usually, in these analyses the original signal is assumed to have been sampled and be in a vector form suitable for transformation by computer processing. Fast digital computers make it conceivable to perform this generalized spectral analysis in real-time. However, additional insight into this signal processing may be obtained if the computer operations are related to more familiar concepts such as frequency filtering.

This paper relates linear transformations to generalized sampling [6] and shows that the computer operations in Hadamard and Haar transformations have an equivalent linear system interpretation. The transformed quantities correspond to sample values of the signal at the output of linear filters that have the amplitude and phase responses of Walsh filters [7]. A generalized sampling expansion can be found that uses these sample values in an expansion that refers the original signal to the output of an arbitrary linear filter.

Generalized Sampling Theory for Stochastic Processes

It is well known that any band-limited function can be expressed in terms of its sample values provided they are taken at a rate that exceeds twice the bandwidth. Such a sampling expansion uses $\sin x/x$ basis functions, and converges in the mean-square sense for band-limited stochastic processes.

The band-limited deterministic signal is known to have a generalized sampling expansion [6]. It may use sample values as well as linear functionals of the signal; as a result, the sampling interval can be increased. It will be shown that this expansion converges in the mean-square sense for band-limited, stationary stochastic processes by bounding the truncation error.

Sampling Theorem

Let $x(t)$ be a real, wide-sense, stationary stochastic process with autocorrelation function denoted by the expectation

$$R(\tau) = E\{x(t + \tau)x(t)\} \quad (1)$$

Its power density spectrum is assumed to be band-limited in the sense that:

$$R(\tau) \leftrightarrow S(\omega) = 0, \quad |\omega| > \omega_0 \quad (2)^*$$

The sampling theorem allows an expansion of $x(t)$ using $\sin x/x$ basis functions if the sampling interval T is less than π/ω_0 . This expansion converges [6] uniformly in the mean-square sense:

$$\lim_{N \rightarrow \infty} E \left\{ \left| x(t) - \sum_{n=-N}^N x(nT) \frac{\sin \omega_0(t-nT)}{\omega_0(t-nT)} \right|^2 \right\} = 0 \quad (3)$$

A block of N samples shall be denoted using vector notation:

$$X(t) = [x(t - (N-1)T), \dots, x(t - T), x(t)]^T \quad (4)$$

A system for deriving this vector signal is shown in Figure 1. N delayed versions of $x(t)$, each sampled once per NT seconds, are used to obtain the block of N samples. This sampling technique is usually assumed to precede any transformation processing.

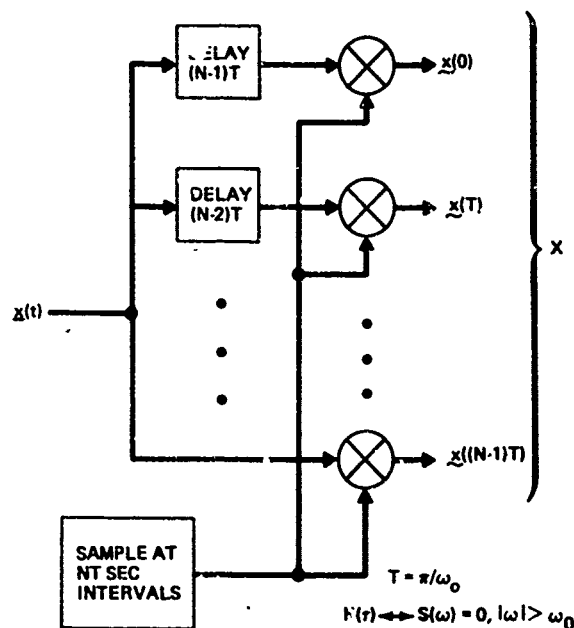


Fig. 1 System for Providing a Vector Signal, X , With Elements That are Sample Values of the Process $x(t)$

* Fourier transform pairs are denoted by a double headed arrow (\leftrightarrow).

Generalized Sampling Theory

Consider the systems of Figure 2. The transfer functions $H_1(\omega), \dots, H_N(\omega)$ are defined for $|\omega| < \omega_0$. Let $Z(\omega)$ be the matrix of these transfer functions with the nm entry given by $H_m[\omega + (n-1)c]$, where $n, m = 1, \dots, N$ and $c = 2\omega_0/N$. If $Z(\omega)$ is non-singular for $-\omega_0 < \omega < \omega_0 + c$, then Papoulis[6] shows that the response of $H(\omega)$ to a band-limited deterministic signal $x(t)$, has a generalized sampling expansion in terms of the sampled values of the responses of $H_1, \dots, H_N(\omega)$ to $x(t)$:

$$g(t + \tau) = \sum_{n=-\infty}^{\infty} \sum_{i=1}^N g_i(t + nT) a_{ni}(\tau), \quad (5)$$

$$T = NT = N\pi/\omega_0$$

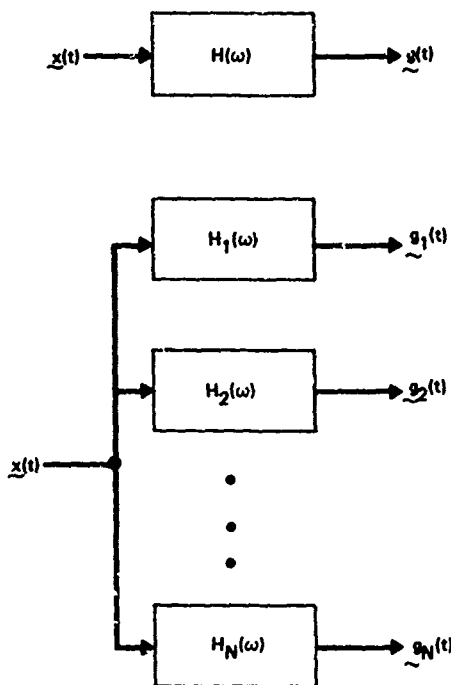


Fig. 2 Generalized Sampling Equivalent Systems

The basis functions used in this expansion are determined from $H(\omega)$, $H_1(\omega)$, \dots , and $H_N(\omega)$. For $Z(\omega)$ non-singular, a set of functions $\{\vartheta_i(\omega)\}$ exists that allows an expansion of the transfer function:

$$H(\omega) e^{j\omega t} = \sum_{i=1}^N H_i(\omega) \vartheta_i(\omega), \quad |\omega| < \omega_0 \quad (6)$$

The $\vartheta_i(\omega)$ are periodic in ω with period c , and are functions of τ . They can be expanded in a Fourier series

$$\vartheta_i(\omega) = \sum_{n=-\infty}^{\infty} a_{ni}(\tau) e^{jnT\omega} \quad (7)$$

The coefficients of this expansion, functions of the variable τ , have been shown to be the set of basis functions for the generalized sampling expansion (5). Their computation is discussed in the next section. For stochastic processes it will be shown that (5) holds in the mean-square sense by establishing bounds on the truncation error

$$e_M(t + \tau) = g(t + \tau) - \sum_{n=-M}^M \sum_{i=1}^N g_i(t + nT) a_{ni}(\tau) \quad (8)$$

Let $\chi(t)$, defined by (1) and (2), be the input to the systems of Figure 2. With $h_i(t) \leftrightarrow H_i(\omega)$ the responses are stochastic processes given by the convolutions

$$g(t) = \chi(t) * h(t) \quad (9)$$

and

$$g_i(t) = \chi(t) * h_i(t), \quad i = 1, 2, \dots, N \quad (10)$$

Formal substitution into (5) yields the output $g(t)$ in terms of the sampled values $g_i(nT)$ of the responses $g_i(t)$. To show that (5) now converges in the mean-square sense, bounds will be established on the truncation error (8), which is now also a stochastic process. Viewing τ as a constant, $e_M(t + \tau)$ is interpreted as the output of a linear system with input $\chi(t)$ and transfer function

$$T(\omega, \tau) = H(\omega) e^{j\omega\tau} - \sum_{n=-M}^M \sum_{i=1}^N a_{ni}(\tau) H_i(\omega) e^{jnT\omega} \quad (11)$$

The power spectrum of the truncation error is given by $S(\omega) |T(\omega, \tau)|^2$; therefore, the mean-square truncation error is given by

$$E \{ e_M^2(t + \tau) \} = \frac{1}{2\pi} \int_{-\omega_0}^{\omega_0} S(\omega) |T(\omega, \tau)|^2 d\omega \quad (12)$$

Substituting (6) into (11) yields

$$T(\omega, \tau) = \sum_{i=1}^N H_i(\omega) \left[\vartheta_i(\omega) - \sum_{n=-M}^M a_{ni}(\tau) e^{jnT\omega} \right] \quad (13)$$

The term in the bracket equals the error in approximating $\vartheta_i(\omega)$ by a truncated Fourier series. If the $\vartheta_i(\omega)$ satisfy known conditions, it follows from the theory of Fourier series that $T(\omega, \tau)$ approaches zero as $M \rightarrow \infty$; hence $\lim_{M \rightarrow \infty} E \{ e_M^2(t + \tau) \} = 0$.

It follows that (5) holds in the mean-square sense for stationary stochastic processes.

Computation of the Reconstruction Functions

The functions used for reconstruction of $g(t + \tau)$ in (5) are the coefficients of the Fourier series for $\theta_i(\omega)$; they are given by

$$a_{ni}(\tau) = \frac{1}{c} \int_{-\omega_0}^{-\omega_0 + c} \theta_i(\omega) e^{-jn\bar{T}\omega} d\omega \quad (14)$$

A matrix formalism will be given for the solution of the $a_{ni}(\tau)$. The $\theta_i(\omega)$ are determined by solving the system of equations

$$\sum_{i=1}^N H_i(\omega + kc) \theta_i(\omega) = H(\omega + kc) e^{j(\omega + kc)\tau} \quad (15)$$

for $k = 0, 1, \dots, N-1$, and $-\omega_0 < \omega < -\omega_0 + c$. The matrix of transfer functions $Z(\omega)$ has been defined. Define a diagonal weighting matrix, $W(\omega)$, the ii entry of which is given by the filter transfer function, $H(\omega)$, evaluated for $-\omega_0 + (i-1)c \leq \omega \leq -\omega_0 + ic$:

$$w_{ii} = H(\omega + (i-1)c) \quad (16)$$

All off-diagonal terms are zero.

Define column vectors of $\theta_i(\omega)$ and exponentials by

$$\Phi(\omega) = [\theta_1(\omega), \dots, \theta_N(\omega)]^t \quad (17)$$

and

$$B(\tau) = [1, e^{jc\tau}, \dots, e^{j(N-1)c\tau}]^t$$

Then (15) can be solved in matrix form for $\theta_i(\omega)$:

$$\Phi(\omega) = e^{j\omega\tau} Z^{-1}(\omega) W(\omega) B(\tau) \quad (18)$$

$\exp(j\omega\tau)$ is a common factor to each $\theta_i(\omega)$. In addition, $\exp(jkc\tau) = \exp(jkc(\tau - n\bar{T}))$, because $kcn\bar{T} = 2kn\pi$; as a result, $a_{ni}(\tau) = a_{0i}(\tau - n\bar{T})$.

Let the column vector of $a_{0i}(\tau)$ be defined by $A_0(\tau) = (a_{01}(\tau), \dots, a_{0N}(\tau))^t$, then

$$A_0(\tau) = \frac{1}{c} \int_{-\omega_0}^{-\omega_0 + c} Z^{-1}(\omega) W(\omega) e^{j\omega\tau} d\omega B(\tau), \quad (19)$$

and the vector of $a_{ni}(\tau)$ is given by

$$A_n(\tau) = A_0(\tau - n\bar{T}) \quad (20)$$

The generalized sampling expansion (5) can be written as:

$$g(t + \tau) = \sum_{n=-\infty}^{\infty} \sum_{i=1}^N a_{0i}(\tau - n\bar{T}) g_i(t + n\bar{T}) \quad (21)$$

Comment

The importance of generalized sampling theory lies in the simplicity of reconstruction, i.e., that a set of functions $\{a_{0i}(\tau)\}$ can be found that allows a sampling expansion which uses sample values that are linear functionals of the original process. For $H(\omega) = 1$ the reconstructed process is the original process, $x(t)$. If the samples $g_i(n\bar{T})$ are quantized, introducing quantization error $q_i(n\bar{T})$ into each sample used to construct the estimate of $x(t)$, then the quantization error in the generalized sampling expansion takes the form (21)

$$e_q(t + \tau) = \sum_{n=-\infty}^{\infty} \sum_{i=1}^N a_{0i}(\tau - n\bar{T}) q_i(t + n\bar{T}) \quad (22)$$

The $a_{0i}(\tau)$ are obtained from (19) with $W(\omega)$ the identity matrix.

In some cases, a more meaningful measure of error results if (22) is weighted, i.e., observed at the output of a linear filter with impulse response $h(t) \leftrightarrow H(\omega)$ (Note $H(\omega)$ is now not necessarily 1, as above). The weighted error is then given by the convolution $e_q(t + \tau) * h(\tau)$. This is equivalent to having used the generalized sampling expansion to construct $h(t) * x(t)$, rather than $x(t)$, with the samples $g_i(n\bar{T})$. The functions $a_{0i}(\tau)$ now obtained from (19) will include the weighting of the diagonal matrix $W(\omega)$, whose entries are obtained from $H(\omega)$. If the samples are quantized, the generalized sampling expansion for the error again takes the form (22); however, this expression now refers the quantization error to the output of the weighting filter.

Generalized Sampling Interpretation of Linear Transformations

Linear transformation of blocks of samples (4) has an equivalent linear system and generalized sampling interpretation. That equivalence is presented and applied to Hadamard and Haar transformations.

Linear Transformation of Blocks of Samples

Let the block of N samples (4) of the process $x(t)$ be transformed by the $N \times N$ matrix H yielding the vector

$$Y = HX \quad (23)$$

The elements of Y are given by

$$y_k = \sum_{i=1}^N h_{ki} x(iT) \quad (24)$$

where $k = 1, 2, \dots, N$, and h_{ki} are the elements of H . Equation (24) has the following interesting interpretation. If

$$g_k(t) = \sum_{i=1}^N h_{ki} x(t - NT + iT) \quad (25)$$

then

$$y_k = g_k(NT) \quad (26)$$

That is, y_k is the sample value, taken every NT seconds, of the response to $x(t)$ of the filter with transfer function

$$H_k(\omega) = \sum_{i=1}^N h_{ki} e^{-j\omega(N-i)T} \quad (27)$$

A block diagram for this interpretation is shown in Figure 3. Each transfer function is the sum of weighted delays of $x(t)$, and obviously has the period $2\omega_0$; however, since $x(t)$ is band limited, each $H_k(\omega)$ may be truncated for $|\omega| > \omega_0$.

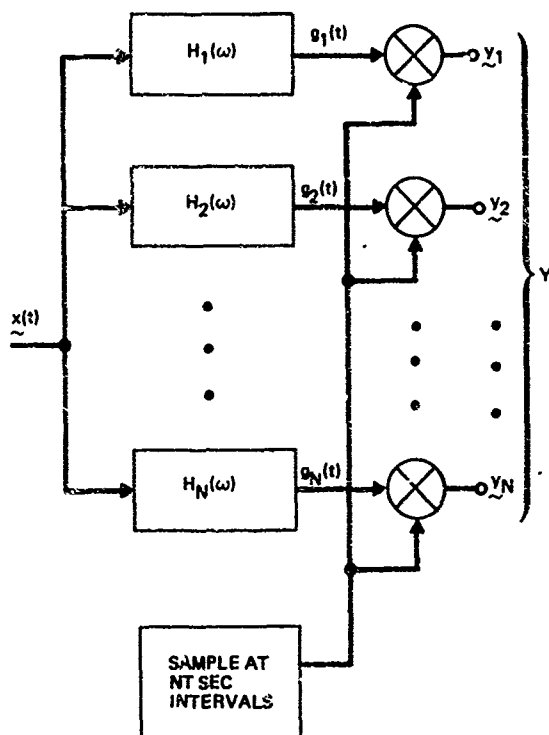


Fig. 3 Generalized Sampling Equivalent to the Linear Transformation $Y=HX$

The system of Figure 3 represents a generalized sampling system if the matrix of transfer functions, $Z(\omega)$, is non-singular. To show that H alone determines this condition, consider the matrix $E(\omega)$ defined as having for its nm entry

$$e_{nm} = \exp(-j(\omega + (n-1)c)(N-m)T) \quad (28)$$

where $n, m = 1, 2, \dots, N$. This is the matrix of transfer functions that corresponds to conventional block sampling (Figure 1). It can be shown that the matrix $E(\omega)/N$ is unitary; hence, the determinant of $E(\omega)$ cannot vanish. The inverse of $E(\omega)$ is given by $(1/N) E^*(\omega)$ (the asterisk signifies matrix adjoint).

From (27) and (28), it can be seen that the matrix of transfer functions $Z(\omega)$ is given by the matrix product

$$Z(\omega) = E(\omega) H^t \quad (29)$$

Its determinant is the product of the determinants of $E(\omega)$ and H . The determinate of $E(\omega)$ cannot vanish; therefore, the non-singularity of $Z(\omega)$ is determined by H . If H is non-singular, a set of N functions $a_{0i}(\tau)$ can be found that allows a sampling expansion (21) using the transformed samples y_k .

The functions used in the generalized sampling expansion (21) can be determined from (19) and (29):

$$A_0(\tau) = \frac{1}{Nc} (H^t)^{-1} \int_{-\omega_0}^{\omega_0} E^*(\omega) W(\omega) e^{j\omega\tau} d\omega B(\tau) \quad (30)$$

$W(\omega)$ (see (16)) is independent of the transformation H and of the $H_i(\omega)$; it represents a filtering or weighting applied to the signal. As a result, the generalized sampling expansion (21) is the sampled process referred to weighting filter output as discussed in the previous section. If the samples are quantized the weighted quantization error can be obtained from the sampling expansion directly.

Application to Hadamard Transform

When the transformation in (23) is a Hadamard matrix, normalized so that each entry is $\pm 1/\sqrt{N}$, the filter responses of (27) are those of Walsh filters[7]. The transfer functions for $N = 2, 4$, and 8 , are given in Table 1 and sketched in Figures 4, 5, and 6. Figure 4 shows that processing with $N = 2$ is equivalent to low- and high-pass filtering of $x(t)$. The filter responses for $N = 4$ are sketched in Figure 5. In addition to low- and high-pass filtering, there is now also some band-pass filtering. For $N = 8$, the filter responses are shown in Figure 6.

Comparison of Figures 4, 5, and 6 shows how the band-pass characteristics of each equivalent filter changes as the block sample size in the processing is increased. Each transformed variable still derives its energy from the total signal spectrum; however, there appears to be increasing frequency selection. As a result, sequence filtering, i.e., setting $y_1 = 0$,

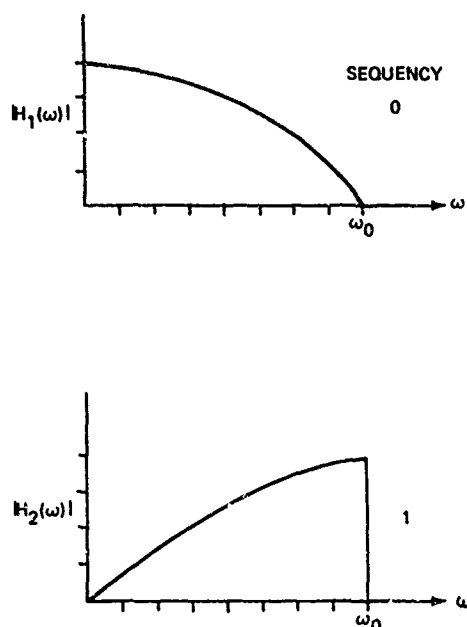


Fig. 4 Frequency Response of Walsh Filters for $N=2$. These Have Been Cut-off at the Limit of the Process (ω_0)

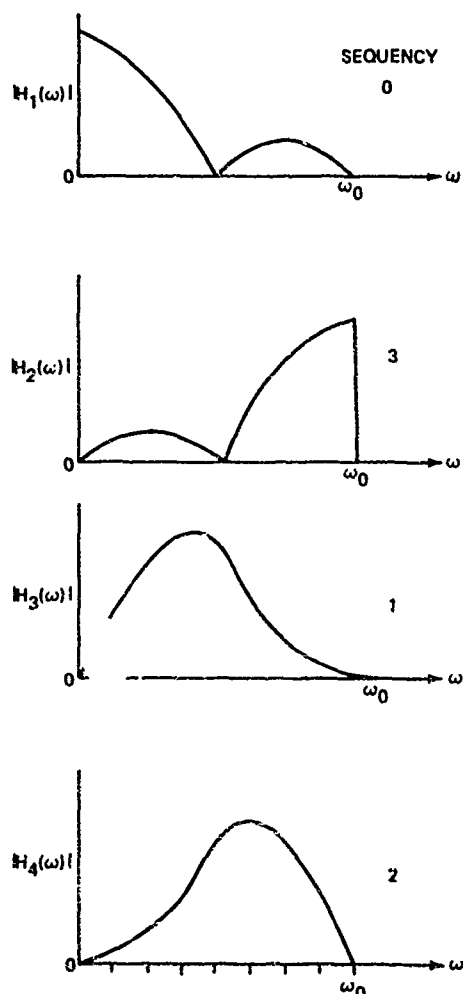


Fig. 5 Frequency Response of Walsh Filters for $N=4$

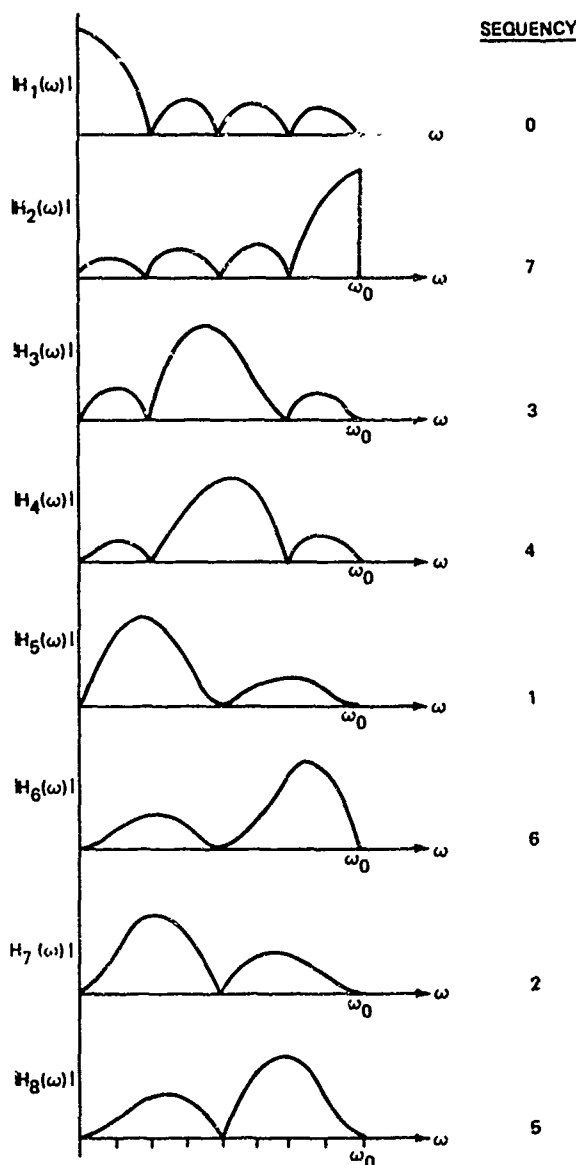


Fig. 6 Frequency response of Walsh Filters for $N=8$

can cause a disproportionate loss of energy in a particular part of the signal spectrum.

Equation (30) has been solved for Hadamard transform and no weighting, i.e., $W(\omega) = 1_N$. The Hadamard matrix is symmetric and orthogonal; therefore, $(H^t)^{-1} = H$. It follows that the functions (30) used in the generalized sampling expansion (21) are linear combinations of the original $\sin x/x$ basis functions:

$$A_0(\tau) = H \left[\sin \omega_0 (\tau + (N-1)T) / (\tau - (N-1)T), \dots, \sin \omega_0 \tau / \omega_0 \tau \right]^t \quad (31)$$

As a trivial example, (31) has been solved for $N = 2$. $x(t)$ has the generalized sampling expansion

$$x(t) = \sum_{n=-\infty}^{\infty} \left\{ \frac{y_0(2nT)}{\sqrt{2}} \left[\frac{\sin \omega_0(t-2nT)}{\omega_0(t-2nT)} + \frac{\sin \omega_0(t-(2n-1)T)}{\omega_0(t-(2n-1)T)} \right] + \frac{y_1(2nT)}{\sqrt{2}} \left[\frac{\sin \omega_0(t-(2n-1)T)}{\omega_0(t-(2n-1)T)} - \frac{\sin \omega_0(t-2nT)}{\omega_0(t-2nT)} \right] \right\}$$

which can be shown to be identical to the sampling expansion. Thus, a new set of basis functions has been found that consists of linear combinations of the original $\sin x/x$ basis.

Application to Haar Transform

An orthogonal sequence of functions, defined by Haar in 1909 is given by:

$$x_0^{(0)}(\theta) = 1$$

$$x_n^{(k)}(\theta) = \begin{cases} 2^{n/2}, \frac{k-1}{2^n} < \theta \leq \frac{k}{2^n} \\ -2^{n/2}, \frac{k-1/2}{2^n} < \theta \leq \frac{k}{2^n} \\ 0 \text{ all other } \theta \text{ in } [0, 1], \end{cases}$$

for $0 \leq \theta < 1$, $n \geq 0$ and $1 \leq k \leq 2^n$. These functions are discussed by Nagy[8] and Andrews[9]. The discrete Haar functions are obtained by sampling continuous Haar functions and are used to write the Haar matrix. The Haar matrix for $n = 0$ is the Hadamard matrix of order 2. Its equivalent filter transfer functions are listed in Table I (under $N = 2$) and shown in Figure 4.

For $n = 1$ the Haar matrix is

$$F_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}$$

The equivalent filter transfer functions are obtained using results for Hadamard filters of order 4 and 2. The first two filters (sequence of 0 and 1) are the same as those for Hadamard transform of order 4. The remaining filters correspond to the high-pass filter (suitably delayed) of the $N = 2$ Hadamard transform. The transfer functions are:

$$H_1(\omega) = 2 \cos \omega T/2 \cos \omega T e^{-j3\omega T/2}$$

$$H_2(\omega) = 2 \cos \omega T/2 \sin \omega T e^{-j(3\omega T/2 + \pi/2)}$$

$$H_3(\omega) = \sqrt{2} \sin \omega T/2 e^{-j(5\omega T/2 + \pi/2)}$$

$$H_4(\omega) = \sqrt{2} \sin \omega T/2 e^{-j(\omega T/2 + \pi/2)}$$

For $n = 2$, the Haar matrix is given by

$$F_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}$$

These transfer functions are again obtained using previous equations. The first two are the same as for Hadamard transform of order 8, sequence 0 and 1 respectively.

$$H_1(\omega) = 2\sqrt{2} \cos \omega T/2 \cos \omega T \cos 2\omega T e^{-j(7\omega T/2)}$$

$$H_2(\omega) = 2\sqrt{2} \cos \omega T/2 \cos \omega T \sin 2\omega T e^{-j(7\omega T/2 + \pi/2)}$$

These transfer functions are shown in Figures 6 and 7. The next two transfer functions are obtained from Hadamard of order 4, but with θ and T both increased by a factor of 2 and each suitably delayed:

$$H_3(\omega) = 2 \cos \omega T/2 \sin \omega T e^{-j(11\omega T/2 + \pi/2)}$$

$$H_4(\omega) = 2 \cos \omega T/2 \sin \omega T e^{-j(3\omega T/2 + \pi/2)}$$

These transfer functions are shown in Figure 5. The last four transfer functions ($m = 5, 6, 7, 8$) are obtained from Hadamard of order 2:

$$H_m(\omega) = \sqrt{2} \sin \omega T/2 e^{-j(\omega T/2 + 2(8-m)\omega T + \pi/2)}$$

These transfer functions are shown in Figure 4. Figure 7 shows all the equivalent Haar filter frequency responses for $n = 2$ and $N = 8$. There is considerable overlapping of the filter pass-bands.

The Haar matrix as defined above has the property $F_n F_n^t = N I_N$, where $N = 2^{(n+1)}$. For the special case of no weighting, $W(\omega) = 1_N$, the basis functions for the generalized sampling expansion (21) are given by

$$A_0(\tau) = \frac{1}{N} F_n^t \left[\sin \omega_0(\tau + (N-1)T) / \omega_0(\tau + (N-1)T), \dots, \sin \omega_0 \tau / \omega_0 \tau \right]^t$$

Conclusions

The generalized sampling expansion was shown to converge in the mean-square sense. It was shown that linear transformation of blocks of sampled data has an equivalent linear system and generalized sampling interpretation. If the transformed samples are quantized, the generalized sampling expansion

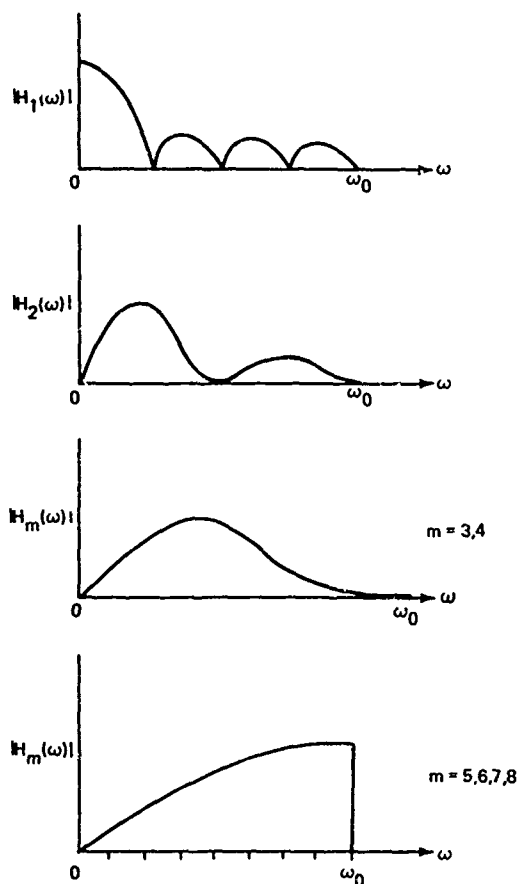


Fig. 7 Haar Filter Frequency Response for $n=2$ and $N=8$

can be used to describe the weighted quantization error in the reconstructed process. For Hadamard and Haar transforms it was shown that the equivalent generalized sampling filter transfer functions are those of the Walsh filters.

Acknowledgement

The author wishes to thank Professor A. Papoulis, of the Polytechnic Institute of Brooklyn, for his guidance and advice in the preparation of this paper.

References

1. H. C. Andrews, "Walsh Functions in Image Processing, Feature Selection and Pattern Recognition," Proceedings of 1971 Symposium on Applications of Walsh Functions (AD-727-000).
2. H. P. Kramer and M. V. Mathews, "A Linear Coding for Transmitting a Set of Correlated Signals," IRE Transactions on Information Theory, IT-2 September 1956, pp 41-46.
3. S. J. Campanella and G. S. Robinson, "A comparison of Walsh and Fourier Transformations for Application to Speech," Proceedings of 1971 Symposium of Applications of Walsh Functions (AD-727-000).
4. D. E. Bowyer, "Walsh Functions, Hadamard Matrices and Data Compression," Proceedings of 1971 Symposium for Applications of Walsh Functions (AD-727-000).
5. L. M. Goodman, "A Binary Linear Transformation for Redundancy Reduction," Proc. IEEE (Letters) March 1967, pp 467, 468.
6. A. Papoulis, Systems and Transforms with Applications in Optics, McGraw-Hill (1968) New York, pp 132-137, 284.
7. G. S. Robinson and R. Granger, "A Design Procedure for Non-Recursive Digital Filters Based on Walsh Functions," 1971 Symposium on Applications of Walsh Functions (AD-727-000).
8. B. Sz-Nagy, Introduction to Real Functions and Orthogonal Expansions, Oxford University Press, New York (1965), pp 338-340.
9. H. C. Andrews and K. L. Caspari, "A Generalized Technique for Spectral Analysis," IEEE Transactions on Computers, January 1970, pp 16-25.

TABLE 1 HADAMARD FILTER TRANSFER FUNCTIONS

Sequency	N = 2	N = 4	N = 8
0	$\sqrt{2} \cos \omega T/2$ $\theta(\omega) = -\omega T/2$	$2 \cos \omega T/2 \cos \omega T$ $\theta(\omega) = -3\omega T/2$	$2\sqrt{2} \cos \omega T/2 \cos \omega T \cos 2\omega T$ $\theta(\omega) = -7\omega T/2$
1	$\sqrt{2} \sin \omega T/2$ $\theta(\omega) = -(\omega T/2 + \pi/2)$	$2 \cos \omega T/2 \sin \omega T$ $\theta(\omega) = -3\omega T/2 - \pi/2$	$2\sqrt{2} \cos \omega T/2 \cos \omega T \sin 2\omega T$ $\theta(\omega) = -7\omega T/2 - \pi/2$
2		$2 \sin \omega T/2 \sin \omega T$ $\theta(\omega) = -3\omega T/2 - \pi$	$2\sqrt{2} \cos \omega T/2 \sin \omega T \sin 2\omega T$ $\theta(\omega) = -7\omega T/2 + \pi$
3		$2 \sin \omega T/2 \cos \omega T$ $\theta(\omega) = -3\omega T/2 - \pi/2$	$2\sqrt{2} \cos \omega T/2 \sin \omega T \cos 2\omega T$ $\theta(\omega) = -7\omega T/2 - \pi/2$
4			$2\sqrt{2} \sin \omega T/2 \sin \omega T \cos 2\omega T$ $\theta(\omega) = -7\omega T/2 + \pi$
5			$2\sqrt{2} \sin \omega T/2 \sin \omega T \sin 2\omega T$ $\theta(\omega) = -7\omega T/2 + \pi/2$
6			$2\sqrt{2} \sin \omega T/2 \cos \omega T \sin 2\omega T$ $\theta(\omega) = -7\omega T/2 + \pi$
7			$2\sqrt{2} \sin \omega T/2 \cos \omega T \cos 2\omega T$ $\theta(\omega) = -7\omega T/2 + \pi/2$

WALSH FUNCTIONS AND THE SAMPLING PRINCIPLE

By

Mohammad Maqusi
Electrical Engineering Department
New Mexico State University
Las Cruces, New Mexico

ABSTRACT

This paper deals with the derivation of the sampling principle using the Walsh analysis techniques. A result analogous to that using Fourier analysis is the consequence. First a generalization of Walsh functions and transforms along with a summary of their useful properties is presented. Then a definition of a delta function to suit the Walsh analysis is established. Finally, the derivation of the sampling principle utilizing Walsh functions is treated.

I. GENERALIZED WALSH FUNCTIONS

Assume that the non-negative real numbers x and z have the dyadic representations:

$$x = \sum_{i=-N}^{\infty} 2^{-i} x_i, \quad z = \sum_{i=-N}^{\infty} 2^{-i} z_i \quad (1)$$

where x_i and $z_i \in \{0,1\}$. Define the addition \oplus and multiplication \otimes operations respectively by:

$$u = x \oplus z = \sum_{i=-N}^{\infty} 2^{-i} u_i,$$

$$u_i = \begin{cases} 0, & \text{if } x_i + z_i \text{ is even} \\ 1, & \text{if } x_i + z_i \text{ is odd} \end{cases} \quad (2)$$

$$w = x \otimes z = \sum_{i=-N}^{\infty} 2^{-i} w_i,$$

$$w_i = \begin{cases} 0, & \text{if } \sum_{K+j=i} x_K z_j \text{ is even} \\ 1, & \text{if } \sum_{K+j=i} x_K z_j \text{ is odd} \end{cases} \quad (3)$$

The notation $\sum_{K+j=i}$ denotes a sum over all possible K and j . $K = -N, -N+1, \dots; j = -N, -N+1, \dots$ such that the condition $K+j=i$ is fulfilled. Utilizing definitions (2) and (3), the generalized Walsh functions are defined as:

$$\psi(z, x) = \psi(1, x \otimes z), \text{ for all } z > 0 \quad (4)$$

where $\psi(\cdot, x)$ denotes the first Walsh function.

Generalized Transform: A generalized Walsh transform is defined with the aid of the functions $\psi(z, x)$ for a function $f(x)$ ($f(x)$ is assumed to be square-integrable) as:

$$W[f(x)] = F(z) = \int_0^{\infty} f(x) \psi(z, x) dx \quad (5)$$

And the inverse transform is subsequently given by:

$$W^{-1}[F(z)] = f(x) = \int_0^{\infty} F(z) \psi(z, x) dz \quad (6)$$

The parameter z is usually referred to as sequency and has units of zeros per second (zps) as it denotes the average number of zero crossings of the function $\psi(z, x)$ in a unit interval.

Properties of the Transform: Some useful properties of the Walsh functions and transforms are stated briefly:

- (1). Orthonormality Principle: the Walsh functions $\psi(n, x)$ form a complete orthonormal set with the property:

$$\int_0^{\infty} \psi(n, x) \psi(s, x) dx = \int_0^{\infty} \psi(n \oplus s, x) dx = \delta_{ns} \quad (\text{Kronecker's delta})$$

$$\delta_{ns} = \begin{cases} 1, & \text{if } n = s \\ 0, & \text{if } n \neq s \end{cases} \quad (7)$$

The parameters n and s are integers.

- (2). Parseval's Theorem: for two arbitrary functions $f(x)$ and $g(x)$, the following relation holds:

$$\int_0^{\infty} f(x) g(x) dx = \int_0^{\infty} F(z) G(z) dz \quad (8)$$

where $F(z) = W[f(x)]$, and $G(z) = W[g(x)]$.

- (3). Shifting Theorem: if $f(x)$ is Walsh-transformed into $F(z)$, then the dyadically-translated function $f(x \oplus a)$ is transformed as:

$$W[f(x \oplus a)] = \psi(z, a) F(z). \quad (9)$$

- (4). Convolution Theorem: if $f(x)$ and $g(x)$ are dyadically-convolved, i.e.

$$h(x) = f(x) \oplus g(x) \quad (10)$$

or

$$h(x) = \int_0^\infty f(t) g(t \oplus x) dt \quad (11)$$

where the operator \oplus denotes dyadic convolution and is defined by (11); then

$$H(z) = F(z) G(z) \quad (12)$$

- (5). Symmetry: from the definition of the functions $\psi(z, x)$ it follows that the functions $\psi(z, x)$ are symmetric in the sense that the parameters z and x are completely interchangeable in any relation involving Walsh functions or transforms, i.e.

$$\psi(z, x) = \psi(x, z) \quad (13)$$

II. DELTA FUNCTION

A delta function is now defined for Walsh functions as:

$$\delta(z) \triangleq \int_0^\infty \psi(z, x) dx \quad (14)$$

i.e. the following transform pairs are obtained

$$f(x) = 1 \leftrightarrow \delta(z) \quad (15a)$$

$$F(z) = 1 \leftrightarrow \delta(x) \quad (15b)$$

where the two-edged arrow denotes a compatible transform pair.

The delta function as defined by (14) is easily extended to the general case of $\delta(z \oplus k)$ by rewriting (14) as:

$$\begin{aligned} \delta(z \oplus k) &= \int_0^\infty \psi(z \oplus k, x) dx \\ &= \int_0^\infty \psi(z, x) \psi(k, x) dx \end{aligned} \quad (16)$$

The sampling property of this function is given in the following theorem.

Theorem I: If $f(t)$ is continuous at a , then

$$\int_0^\infty f(t) \delta(t \oplus a) dt = f(a) \quad (17)$$

Proof: Let $f(t)$ and $F(z)$ be a Walsh transform pair. From (16) the left-hand side of (17) can be written as

$$\begin{aligned} &\int_0^\infty f(t) \delta(t \oplus a) dt \\ &= \int_0^\infty f(t) \left[\int_0^\infty \psi(t, x) \psi(a, x) dx \right] dt \end{aligned} \quad (17a)$$

Interchanging the order of integration (i.e. integrating w.r. to t first) gives

$$\begin{aligned} &\int_0^\infty \psi(a, x) \left[\int_0^\infty f(t) \psi(t, x) dt \right] dx \\ &= \int_0^\infty \psi(a, x) F(x) dx = f(a) \end{aligned} \quad (17b)$$

From the theorem it is easily seen that

$$\int_0^\infty \delta(t \oplus a) dt = 1 \quad (18)$$

Relations (17) and (18) ascribe an interpretation to the delta function as it is used in Fourier analysis.

III. SAMPLING PRINCIPLE

Theorem II: Assume that $f(t)$ is a sequence bandlimited signal, with bandwidth $2B$ zps in the frequency domain. If $f(t)$ is sampled at intervals of $1/2B$ sec., then it is possible to reconstruct $f(t)$ from the knowledge of these samples.

Proof: With $f(t)$ restricted as in the theorem, the following constraint is obtained

$$F(k) = 0, \text{ for all } k < 0 \text{ and } k > 2B \quad (19)$$

And $F(k)$ can be extended to yield a periodic function of period $2B$. This periodic function can subsequently be expanded in a Walsh series as:

$$F(k) = \sum_{n=0}^\infty A(n) \psi(n, k), \quad 0 < k < 2B \quad (20)$$

where the expansion coefficients $A(n)$ are evaluated by,

$$A(n) = \frac{1}{2B} \int_0^{2B} F(k) \psi(n, k) dk \quad (21)$$

Now,

$$f(x) = \int_0^{2B} F(k) \psi(k, x) dk \quad (22)$$

Using the symmetric property of Walsh functions, i.e.

$$\psi(k, x) = \psi(x, k)$$

gives

$$f(x) = \int_0^{2B} F(k) \psi(x, k) dk \quad (23)$$

Evaluating $f(x)$ at the sampling instants $x_n = n/2B$ yields

$$f(x_n) = f(\frac{n}{2B}) = \int_0^{2B} F(k) \psi(\frac{n}{2B}, k) dk \quad (24)$$

or

$$f(x_n) = 2B A(n/2B) \quad (25)$$

Thus, it is concluded from (25) that the expansion coefficients $A(n)$ can be derived from the values of $f(x)$ at the sampling instants $x = 0, 1/2B, 2/2B, 3/2B, \dots$. And, $F(k)$ can be determined uniquely by the values of the sampled ordinates as follows:

$$F(k) = \sum_{n=0}^{\infty} A(n) \psi(n, k), \quad 0 < k < 2B \quad (26a)$$

or

$$F(k) = \sum_{n=0}^{\infty} \frac{1}{2B} f(\frac{n}{2B}) \psi(n, k) \quad (26b)$$

Hence, knowledge of $F(k)$ enables the determination of $f(x)$ for all possible x , i.e.

$$f(x) = \int_0^{2B} F(k) \psi(k, x) dk \quad (27a)$$

or

$$f(x) = \int_0^{2B} \frac{1}{2B} \left[\sum_{n=0}^{\infty} f(\frac{n}{2B}) \psi(n, k) \right] \psi(k, x) dk \quad (27b)$$

Interchanging the order of integration and summation in (27b) gives

$$f(x) = \frac{1}{2B} \sum_{n=0}^{\infty} f(\frac{n}{2B}) \int_0^{2B} \psi(n, k) \psi(k, x) dk \quad (28)$$

Let the integral in (28) be designated as:

$$I(m, B) = \int_0^{2B} \psi(n, k) \psi(k, x) dk, \quad m = n \oplus x, \quad (29)$$

Using the symmetric property of Walsh functions as,

$$\psi(k, x) = \psi(x, k) \quad (30)$$

yields

$$I(m, B) = \int_0^{2B} \psi(n, k) \psi(x, k) dk = \int_0^{2B} \psi(m, k) dk, \quad (31)$$

Fine discussed a similar integral defined as

$$J(m, x) = \int_0^x \psi(m, t) dt, \quad m=0, 1, 2, \dots \quad (32a)$$

The result for a general index m is given by

$$J(m, x) = 2^{-(n+2)} [\psi(m', x) - \sum_{r=1}^{\infty} 2^{-r} \psi(2^{n+r} + m, x)] \quad (32b)$$

where for $m \geq 1$, m can be written as $m = 2^n + m'$, and $0 < m' < 2^n$.

Define the integral

$$D(k, B) = \int_0^B \psi(k, x) dx \quad k \text{ is real}, \quad (33)$$

Using the definition of generalized Walsh functions as

$$\psi(k, x) = \psi(k_0, x) \psi(x_0, k) \quad (34)$$

where k_0 and x_0 denote the greatest integers in k and x respectively.

Thus,

$$D(k, B) = \int_0^B \psi(k_0, x) \psi(x_0, k) dx \quad (35)$$

Two cases concerning (35) are analyzed separately.

1) $B < 1.0$.

This yields

$$D(k, B) = \int_0^B \psi(k_0, x) dx \quad (36)$$

which in terms of Fine's integral is evaluated as

$$D(k, B) = J_{k_0} (B) \quad (37)$$

2) $B > 1.0$.

In this case the result of (35) is simplified by dividing the $(0, B)$ interval into integral subintervals of unit length.

Hence, in the $(p, p+1)$ subinterval

$$x_0 = p, \quad p \leq x < p+1 \quad (38)$$

Thus

$$D(k, B) = \sum_{p=0}^{B_0-1} \psi(p, k) \int_p^{p+1} \psi(k_0, x) dx + \psi(B_0, k) \left[\int_{B_0}^B \psi(k_0, x) dx \right] \quad (39)$$

The last integral in (39) covers the end sub-interval which might differ from unit length. And so in terms of Fine's J-function, (39) simplifies to

$$D(k, B) = \sum_{p=0}^{B_0-1} \psi(p, k) [j_{k_0}(p+1) - j_{k_0}(p)] + \psi(B_0, k) [j_{k_0}(B) - j_{k_0}(B_0)] . \quad (40)$$

Returning to the original integral in (31),

$$I(m, b) = \int_0^{2B} \psi(m, k) dk, \quad m \text{ in real}, \quad (41)$$

Hence

$$I(m, b) = D(m, 2B) \quad (42)$$

And (28) reduces to

$$f(x) \sim \frac{1}{2B} \sum_{n=0}^{\infty} f\left(\frac{n}{2B}\right) D(m, 2B), \quad m = n \oplus x, \quad (43)$$

Eq. (43) clearly indicates that the function $f(x)$ can be constructed from its sample values at

$x_n = \frac{n}{2B}$, which is the conclusion of the sampling principle.

IV. CONCLUSION

The desired result of sampling principle is manifested in (43). This result formulates the principle in the x -domain, and a corresponding result can be obtained in the sequence (z -) domain by following analogous steps of derivation.

The derivation of (43) for Walsh analysis is seen to parallel the derivation of sampling principle in the classical Fourier analysis.

ACKNOWLEDGMENT

The author would like to thank Dr. L. C. Ludeman for his suggestions concerning the paper.

REFERENCES

- Crittenden, R. B. (1970), Walsh-Fourier Transforms, 1970 Symposium on Applications of Walsh Functions, Naval Research Lab., Washington, D. C.
- Fine, N. J. (1949), On the Walsh Functions, Trans. Am. Math. Soc., Vol. 65, pp. 372-414.
- Panter, F. P. (1965), "Modulation, Noise and Spectral Analysis", Chap. 17, McGraw-Hill, New York.
- Pearl, J. (1971), Application of Walsh Transform to Statistical Analysis, IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-1, No. 2, pp. 111-119.

SAMPLING EXPANSIONS IN DISCRETE AND FINITE

WALSH-FOURIER ANALYSIS

Chon Tam Le Dinh, Graduate Student
Phung Le, Graduate Student
Roger Goulet, Assistant Professor
Digital Transmissions Group
Department of Electrical Engineering,
University of Sherbrooke, Sherbrooke, Canada.

Introduction

The problem of sampling expansion in the case of continuous functions has been studied recently [1]. But digital transmission, high speed digital computer image processing and pattern recognition techniques deal with large sets of finite and discrete signals instead of classical continuous and infinite functions. Therefore, it is desirable to review the above problem in the case of discrete and finite functions.

Section II studies some properties of truncated Walsh-Fourier (WF) finite series. Characteristics of partial sums and their transforms are pointed out. Orthogonality of WF kernel is proved. Interpretation of dyadic derivative is given in terms of Féjer sum. In Section III, the concepts of "M-sequence band-limited" signal, (MBL), and "M-sequence band-pass" signal, (MBP) are introduced. From these concepts, sampling expansion theorems are shown. It is proved that a MBL or a MBP signal can be recovered only by $M = 2^m$ of their values, properly chosen. In Section IV, various bounds involving the MBL signals are estimated using the technique based on maximum response of linear dyadic system.

Definitions and Notations

The functions to be considered in this paper are defined on a domain B_n , the set of non negative integers less than $N = 2^n$, where n in turn is a positive integer. Each element x of B_n has a unique expansion

$$x = \sum_{r=0}^{n-1} x_r 2^r \quad (1)$$

with n coefficients in $\{0,1\}$.

The discrete Walsh-Kaczmarz functions of order n are defined as in [2]

$$\text{wal}(u,x) = \exp \pi i \left[\sum_{r=0}^{n-1} (u_{n-r} \oplus u_{n-r-1}) x_r \right] \quad (2)$$

where the operators \oplus and $\#$ denote dyadic addition, and $u_n = 0$. The variable u is used for denoting the number of zero crossings in the range $0 \leq x < N$. The sequence of $\text{wal}(u,x)$ may be defined as integer part of the half of $u+1$ [2], but in the paper the term sequence is used instead of zero-crossing, if no otherwise specified.

The set $\{\psi(k,x)\}$ where

$$\psi(k,x) = \exp \pi i \sum_{r=0}^{n-1} k_r x_r \quad (3)$$

is called the discrete Walsh-Paley functions [2], and is identical with the rows in the natural ordering of the Hadamard matrix. The one-to-one correspondence between u and k such that $\text{wal}(u,x) = \psi(k,x)$ has been given in [3].

The WF transform of a function $f(x)$ and its inverse transform are defined respectively as follows:

$$F(u) = N^{-1} \sum_{x=0}^{N-1} f(x) \text{wal}(u,x), \quad (4)$$

$$f(x) = \sum_{u=0}^{N-1} F(u) \text{wal}(u,x). \quad (5)$$

Truncated Walsh Finite Series and Féjer Sum

Truncated Walsh Finite Series

Let define the partial sum

$$\hat{f}_p(x) = \sum_{u=0}^{p-1} F(u) \text{wal}(u,x). \quad (6)$$

Substituting Eq.4 into Eq.6, and inverting the summation order lead to

$$\hat{f}_p(x) = \sum_{j=0}^{N-1} \sum_{u=0}^{p-1} \text{wal}(u,j \otimes x) f(j). \quad (7)$$

Let the WF kernel be

$$d_p(x) = \sum_{u=0}^{p-1} \text{wal}(u,x). \quad (8)$$

Substituting Eq.8 into Eq.7

$$\begin{aligned} \hat{f}_p(x) &= N^{-1} \sum_{j=0}^{N-1} d_p(x \otimes j) f(j) \\ &= d_p(x) * f(x) \end{aligned} \quad (9)$$

where $*$ denotes the logical convolution. Eq.9 proves that $\hat{f}_p(x)$ is the weighted average of $f(x)$ with the $d_p(x)$ as weighting function.

Consider now the case where $p=M=2^m$. It is easy to verify that

$$d_M(x) = \sum_{r=0}^{R-1} \delta(x + r) \quad (10)$$

$$\text{where } R = NM^{-1} \quad (10.a)$$

and the discrete delta function [3]

$$\delta(x) = \delta_{x,0} \quad (10.b)$$

The right handside of Eq. 10.b is the Kronecker notation. Using Eq.9 and Eq.10, a simple manipulation gives

$$\hat{f}_M(x) = R^{-1} \sum_{r=0}^{R-1} f(x + r). \quad (11)$$

Now let

$$x = pR + s \quad (12)$$

where $p=0,1,\dots,M-1$ and $s=0,1,\dots,R-1$.

Using Eq.12, Eq.11 can be rearranged as follows

$$\hat{f}_M(x) = R^{-1} \sum_{p=0}^{M-1} \sum_{s=0}^{R-1} \sum_{r=0}^{R-1} \{f(pR + s + r) \delta(x - pR - s)\}.$$

Applying the invariance lemma [2], the last expression becomes

$$\hat{f}_M(x) = \sum_{p=0}^{M-1} \left\{ R^{-1} \sum_{r=0}^{R-1} f(pR + r) \sum_{s=0}^{R-1} \delta(x - pR - s) \right\}. \quad (13)$$

Investigating Eq.13 shows that $\hat{f}_M(x)$ is composed of M trains of impulses, each of these trains is formed by R impulses of which the amplitudes are equal to the mean of R corresponding values of the original signal $f(x)$.

It is easy to verify that the transform of $\hat{f}_M(x)$ can be obtained as

$$\hat{F}_M(u) = F(u) \sum_{j=0}^{M-1} \delta(u + j). \quad (14)$$

From Eq.13 and Eq.14 it is seen that:
Statement 1

a. The set of M mutually exclusive and exhaustive subsums

$$R^{-1} \sum_{r=0}^{R-1} f(pR + r); p = 0,1,\dots,M-1$$

and the set of M first components

$$F(u); u = 0,1,\dots,M-1$$

form a M -Walsh-Fourier transforms pair.

b. By duality, the set of M mutually exclusive and exhaustive subsums

$$R^{-1} \sum_{r=0}^{R-1} F(pR + r); p = 0,1,\dots,M-1$$

and the set of M first values

$$f(x); x = 0,1,\dots,M-1$$

form a M -Walsh-Fourier transforms pair. ∇

It is incidently noted that similar results for Hadamard transforms have been also obtained [4], using different approach.

Orthogonality of the Walsh-Fourier kernel

It is maintained that

∇ Statement 2

a. $\{d_M(x + pR); p = 0,1,\dots,M-1\}$ and

b. $\{d_M(x + q2R), d_M(x + (2R-1) + q2R); q = 0,1,\dots,M-1\}$

are two uncomplete orthogonal sets. ∇

Proof It is first noted that the transform of a WF kernel $d_M(x)$ is

$$D_M(u) = \sum_{j=0}^{M-1} \delta(u + j) \quad (15)$$

Applying the Parseval's theorem

$$\begin{aligned} \sum_{x=0}^{N-1} d_M(x + pR) d_M(x + qR) &= \\ &= N \sum_{u=0}^{N-1} D_M^2(u) \text{wal}(u, pR) \text{wal}(u, qR); \\ &= N \sum_{u=0}^{M-1} \text{wal}(u, pR) \text{wal}(u, qR); \text{ from Eq.8,} \\ &= N d_M((p + q)R); \text{ from Eq.10,} \\ &= \begin{cases} NM & \text{for } (p+q)R=0,\dots,R-1, \text{ ie, } p=q \\ 0 & \text{otherwise.} \end{cases} \quad (16) \end{aligned}$$

Hence the statement 2a. is proven.

The proof of the statement 2b. can be obtained as a replica of the above proof, then omitted.

Féjer Sum and Dyadic Derivative

Let the Féjer sum and the Féjer kernel be defined, respectively as the follows

$$\sigma_q(x) = q^{-1} \sum_{p=1}^q \hat{f}_p(x), \quad (17)$$

$$K_q(x) = q^{-1} \sum_{p=1}^q d_p(x). \quad (18)$$

Substituting Eq.9 into Eq.17 results:

$$\begin{aligned} \sigma_q(x) &= N^{-1} \sum_{j=0}^{N-1} K_q(j + x) f(j) \\ &= K_q(x) * f(x). \quad (19) \end{aligned}$$

Thus, $\sigma_q(x)$ equals the weighted average of $f(x)$ with Féjer kernel as weight.

Furthermore, from Eqs 6 & 17, by a simple calculation

$$\sigma_q(x) = \sum_{u=0}^{q-1} (1 - (u/q)) F(u) \text{wal}(u, x). \quad (20)$$

Let now $q = N$, Eq. 20 can be written as

$$\sum_{u=0}^{N-1} uF(u) \text{wal}(u, x) = N[f(x) - \sigma_N(x)]. \quad (21)$$

The left hand side of Eq. 21 is the logical derivative of the second kind, $f^{(1)}(x)$ as defined in [3]. Then,

Statement 3

$$f^{(1)}(x) = N[f(x) - \sigma_N(x)], \quad (22)$$

in words, at every point $x = 0, 1, \dots, N-1$, the value of the logical derivative of the second kind is equal to N times the error in the approximation of $f(x)$ by the Féjer sum $\sigma_N(x)$. \forall

Finally, combining Eq. 19 with $q = N$ and Eq. 12, and noting [3]

$$N\delta^{(1)}(x) \oplus f(x) = f^{(1)}(x)$$

lead to

$$K_N(x) = N\delta(x) - \delta^{(1)}(x). \quad (23)$$

Eq. 23 expresses the Féjer Kernel in terms of delta function and its dyadic derivative.

Sampling Expansions

Let $C = 2^C$ and $M = 2^M$ and $C, C+M \leq N$ and R be determined by Eq. 10a.

Definition 1

A function $f(x)$ is said M -sequence band-limited signal, (MBL), if its WF transform $F(u)$ is zero for $u \geq M$. \forall

Definition 2

A function $f(x)$ is said M -sequence band-pass signal, (MBP), if its WF transform $F(u)$ is zero outside $C \leq u < C+M$.

More specifically $f(x)$ is said to be

- * a narrow band-pass signal, if $C > M$
- * a wide band-pass signal, if $C < M$.

The value M is called sequence band-width of $f(x)$. C is the cutoff low frequency, $C+M$ is the cutoff high frequency. \forall

Various Forms of Sampling Expansions for MBL signals

It is maintained that if $f(x)$ is a MBL signal then $f(x)$ can be reconstructed completely from its M values:

Statement 4

$$f(x) = M^{-1} \sum_{p=0}^{M-1} f(pR) d_M(x \oplus pR). \quad (24) \forall$$

Statement 5

$$f(x) = M^{-1} \sum_{p=0}^{M-1} f(pR \oplus a) d_M(x \oplus pR \oplus a) \quad (25)$$

where a is an arbitrary integer in B_N . \forall

Statement 6

$$f(x) = M^{-1} \sum_{p=0}^{M-1} f(p2R) d_M(x \oplus p2R)$$

$$+ M^{-1} \sum_{p=0}^{M-1} f(p2R \oplus (2R-1)) d_M(x \oplus (2R-1) \oplus p2R) \quad (26) \forall$$

Investigating Eqs. 24, 25 & 26 the following comments can be obtained.

Eq. 25 is a general form of Eq. 24. Applications of Eq. 25 could be useful in signal multiplexing.

Using Eqs. 24 and/or 25 the M sampling points are equidistantly spaced on the x -axis, while for Eq. 26 the sets of M sampling points is divided into two groups, each of these groups is formed by $\frac{1}{2}M$ sampling points regularly distributed on the x -axis. These different results are obvious in the light of Statement 2. The sampling points are illustrated in Fig. 1 with $N=16$ and $M=4$.

The proof of Statement 4 is based upon the following lemma.

Lemma

$$R^{-1} \sum_{p=0}^{M-1} \text{wal}(pM, x) = R \sum_{p=0}^{M-1} \delta(x \oplus p2R) \quad (27)$$

$$+ R \sum_{p=0}^{M-1} \delta(x \oplus (2R-1) \oplus p2R). \forall$$

Proof of this lemma is given elsewhere [5].

Proof of Statement 4. The following proof uses the convolution technique.

Let consider the impulse train

$$h(x) = NR \sum_{p=0}^{M-1} \delta(x \oplus pR). \quad (28)$$

Its WF transform is

$$H(u) = R \sum_{p=0}^{M-1} \text{wal}(pR, u). \quad (29)$$

Applying the above lemma, with x, M replaced respectively by u, R ,

$$H(u) = N \sum_{p=0}^{R-1} \delta(u \oplus p2M) + N \sum_{p=0}^{R-1} \delta(u \oplus (2M-1) \oplus p2M). \quad (30)$$

Let $F(u)$ be the WF transform of an arbitrary MBL signal $f(x)$. Consider the logical convolution $F(u) \odot h(u)$, with $H(u)$ given by Eq. 30:

$$F(u) \odot H(u) = \sum_{p=0}^{R-1} F(u \oplus p2M) + \sum_{p=0}^{R-1} F(u \oplus (2M-1) \oplus p2M). \quad (31)$$

Eq. 31 is illustrated in Fig. 2b for $N=16$ and $M=4$. Investigating Fig. 2b suggests

that to regain $F(u)$ it is sufficient to multiply $F(u) \otimes H(u)$ by the function $W(u)$ shown in the Fig.2c:

$$F(u) = \{F(u) \otimes H(u)\} \cdot W(u). \quad (32)$$

Eq.32 can be obtained from Eq.31 by noting that, for $p=0$ and $u=0,1,\dots,M-1$, the second member of Eq.31 is exactly equal to $F(u)$.

The inverse transform of Eq.32 is

$$f(x) = N^{-1} \{f(x) \cdot h(x)\} \otimes w(x). \quad (33)$$

Substituting Eq.28 into Eq.33, and by some manipulations

$$f(x) = M^{-1} \sum_{p=0}^{M-1} f(pR) w(x \oplus pR). \quad (34)$$

From Eq.15, it is clear that

$$w(x) = d_M(x). \quad (35)$$

Hence Eq.24 is proved by substituting Eq.35 into Eq.34.

Proof of Statement 5. Proof of this statement is straightforward by applying Eq.24 to $f(x \otimes a)$ which is a MBL signal, and replacing $x \otimes a$ by x in the sampling expansion of $f(x \otimes a)$.

Proof of Statement 6. Proof of this statement is very similar to that of Statement 4. The only difference is to start with the following function instead of Eq.28.

$$h(x) = NR \sum_{p=0}^{M-1} \{\delta(x \oplus pR) + \delta(x \oplus (2R-1) \oplus p2R)\}$$

whose WF transform is

$$H(u) = N \sum_{p=0}^{R-1} \delta(u \oplus pM).$$

To save space other details of calculation are omitted.

Sampling Expansions for MBP Signals

The WF transform of a MBP signal $f(x)$ of cutoff low frequency C is given by

$$F(u) = \sum_{j=C}^{C+M-1} F(j) \delta(u \otimes j). \quad (36)$$

It is maintained that

Statement 7 If $f(x)$ is a narrow band-pass signal then

$$f(x) = M^{-1} \sum_{p=0}^{M-1} f(pR) d_M(x \oplus pR) \text{wal}(C, x \oplus pR). \quad (37) \nabla$$

Proof

If $C > M$ then $j-C = j \otimes C$ for $j=C, \dots, C+M-1$. Thus E-36 becomes

$$F(u) = \sum_{r=0}^{M-1} F(r \otimes C) \delta(u \otimes r \otimes C). \quad (38)$$

Let define

$$a(x) = f(x) \text{wal}(C, x) \quad (39)$$

or

$$f(x) = a(x) \text{wal}(C, x). \quad (40)$$

From Eq.38, it is easy to see that the WF transform of $a(x)$ is

$$A(u) = F(u \otimes C) = \sum_{r=0}^{M-1} F(r \otimes C) \delta(u \otimes r).$$

$a(x)$ is then a MBL signal. Applying Eq.24 to $a(x)$, and using Eqs. 39 & 40 a simple calculation gives Eq.37, hence Statement 7.

To derive a sampling expansion for a wide band-pass signal $f(x)$ the following definition is first introduced.

Definition 3 The i -th signal of the set of $S = MC^{-1}$ CBL signals associated with $f(x)$ is defined by

$$f_i(x) = f(x) \text{wal}(iC, x) \otimes d_C(x), \quad (41)$$

$i = 1, 2, \dots, S.$ ∇

The set of CBL signals associated with $f(x)$ can be generated by the dyadic system shown in Fig.3.

It is maintained that

Statement 8 If $f(x)$ is a wide band-pass signal then

$$f(x) = C^{-1} \sum_{i=1}^S \sum_{p=0}^{C-1} f_i(pT) \text{wal}(iC, x) d_C(x \oplus pT) \quad (42)$$

where $T = NC^{-1}.$ ∇

Proof For $C < M$, Eq.36 becomes

$$F(u) = \sum_{i=1}^S \sum_{r=0}^{C-1} F(r \otimes iC) \delta(u \otimes r \otimes iC). \quad (43)$$

Investigating Eq.43 shows that $F(u)$ can be considered as the sum of S narrow band-pass signals $a_i(x)$ of bandwidth C .

$$A_i(u) = \sum_{r=0}^{C-1} F(r \otimes iC) \delta(u \otimes r \otimes iC) \quad (44)$$

where $i = 1, 2, \dots, S.$

From Eqs.41 & 44 it is easy to see that the transform $F_i(u)$ of the i -th CBL signal associated with $f(x)$ satisfies

$$A_i(u) = F_i(u \otimes iC). \quad (45)$$

Substituting Eq.45 into Eq.43, then taking the inverse transform gives

$$f(x) = \sum_{i=1}^S f_i(x) \text{wal}(iC, x). \quad (46)$$

Applying Eq.24 for each CBL signal $f_i(x)$, and replacing the results into Eq.46 give Eq.42. Hence Statement 8.

Sampling Expansion for Some Specific Spectrum Signals

In the above statements it has been proved that $f(x)$ is MBL or MBP signal then $f(x)$ can be regained from only M samples, properly chosen. Consider a function $f(x)$ whose Walsh spectrum has only M components, but it cannot be considered neither as a MBL nor as a MBP signal. The following question could be asked. "How one can regain this signal from only M properly chosen samples?" A partial but useful answer can be obtained as follows in the case where the Walsh spectrum has some specific patterns. It is known that there exists a one-to-one correspondence between Walsh and Hadamard functions [3]. Let define three new classes of signals whose Walsh spectrum components are "MBL or MBP" components in the Hadamard domain. Then this above problem could be considered as solved for these new classes of signals, if sampling expansions using Hadamard functions are known. Fig. 4 illustrates this technique for a signal whose Walsh spectrum components are transformable into MBL components in Hadamard domain, for $N = 16$ and $M = 4$.

The above technique can be generalized to cover other specific signals whose Walsh spectrum is transformable into a MBL or MBP spectrum in other domains (provided that there exists a one-to-one mapping between these domains and Walsh domain).

In the following, the sampling expansions using Hadamard functions are considered.

Let the Hadamard kernel be defined as:

$$d_M^H(x) = \sum_{k=0}^{M-1} \psi(k, x). \quad (47)$$

It is claimed that:

Statement 9

The set

$$\{d_M^H(x \oplus p) ; p = 0, 1, \dots, M-1\}$$

is uncomplete orthogonal one. \forall

Statement 10 If $f(x)$ is a Hadamard MBL signal then $f(x)$ can be regained without error using its M first values.

$$f(x) = M^{-1} \sum_{p=0}^{M-1} f(p) d_M^H(x \oplus p). \quad (48) \quad \forall$$

Statement 11 If $f(x)$ is a Hadamard narrow band-pass signal with cutoff parameters C and $C+M$ then

$$f(x) = M^{-1} \sum_{p=0}^{M-1} f(p) d_M^H(x \oplus p) \psi(C, x \oplus p). \quad (49) \quad \forall$$

Statement 12 If $f(x)$ is a Hadamard wide bandpass signal with cutoff parameters C and $C+M$ then

$$f(x) = C^{-1} \sum_{i=1}^S \sum_{p=0}^{C-1} f_i^H(p) \psi(iC, x) d_C^H(x \oplus p \oplus 50)$$

where

$$f_i^H(x) = f(x) \psi(iC, x) \bullet d_C^H(x) \quad (51)$$

is the i -th signal in the set of $S = MC^{-1}$ Hadamard CBL signals associated with $f(x)$. \forall

Proofs of Statements 9-12 can be obtained by use the same techniques given in the foregoing analysis, hence omitted.

It is interesting to note that the WF kernel can be considered as member of a set of interpolation operators, while the Hadamard kernel defined by Eq. 47 as member of a set of extrapolation operators.

Various Bounds Involving MBL Signals

Bounds involving MBL signals can be estimated by using the concept of maximum response [6] of dyadic linear systems. Applying the well-known Schwarz inequality, the following results are easy to be obtained.

Maximum response of Dyadic Linear Systems

Let $f(x)$ be a MBL signal with bounded energy E :

$$E = \sum_{x=0}^{N-1} f^2(x). \quad (52)$$

If $f(x)$ is the input to a dyadic linear system defined by its response $h(x)$ to the impulse $\delta(x)$ or by its system function $H(u)$ then the system output $g(x)$ is upper bounded

$$|g(x)| \leq \left[EN^{-1} \sum_{u=0}^{M-1} h^2(u) \right]^{1/2}. \quad (53)$$

This bound is attained at a predetermined value x_0 only if the input is given as follows:

$$f_0(x) = \alpha h(x \oplus x_0) \bullet d_M^H(x). \quad (54)$$

where the constant α is determined in such a way that the energy of $f_0(x)$ be equal to E .

In the case where $M = N$

$$h(x) = \alpha^{-1} f_0(x \oplus x_0) \quad (55)$$

is called dyadic matched filter for $f_0(x)$.

Signal bound From Eq. 53 it is easy to see that for $H(u) = 1$

$$|f(x)| \leq E^{1/2} R^{-1/2}. \quad (56)$$

Dyadic Derivative Bound Applying Eq. 23 with $H(u) = u$ gives

$$|f^{(1)}(x)| \leq \left[\frac{E(M-1)(2M-1)}{6R} \right]^{1/2}. \quad (57)$$

The above estimated bounds are useful for predicting computer memory in the processing of signal. In the following, the error bound in the approximation of a frequency limited signal by a MBL will be considered.

Aliasing Error

Consider a function $f(x)$ whose WF transform $F(u)$ is zero for $u \geq U$. Suppose that $M < U < 2M+1$. Let $f(x)$ be approximated by the MBL signal

$$f_a(x) = M^{-1} \sum_{p=0}^{M-1} f(pR) d_M(x \oplus pR).$$

The error in this approximation

$$e_a(x) = f(x) - f_a(x) \quad (58)$$

is called aliasing error. It has been proved [5] that $e_a(x)$ is upper bounded

$$|e_a(x)| \leq B |wal(M, x) - wal(M-1, x)| \quad (59)$$

$$\text{where } B = \sum_{u=M}^{N-1} |F(u)|.$$

It is incidently noted that for Hadamard MBL signal approximation, the resulted error is similar

$$|e_a^H(x)| \leq B^H |\psi(M, x) - 1| \quad (60)$$

$$\text{where } B^H = \sum_{k=M}^{N-1} |F^H(k)|.$$

Investigating Eq.59 (or Eq.60) shows that the approximation of $f(x)$ by $f_a(x)$ is exact at $\frac{1}{2}N$ points x such that the expression in the absolute sign of Eq.59 is zero. At the remaining $\frac{1}{2}N$ points the error is upper bounded by $2B$.

Conclusion

The well-known concept of partial sum, Fourier kernel, Féjer sum and Féjer kernel in Fourier analysis, are generalized into Walsh domain for investigating discrete and finite functions. The Féjer sum in Walsh domain leads to an interpretation of the dyadic derivative of the second kind. The WF kernel is proved to be useful in investigating of truncated Walsh finite series. Its orthogonality properties yield various sampling expansions. Using these sampling expansions MBL and MBP signals can be recovered without error using M properly chosen samples. The introduction of the Fourier-Hadamard kernel leads to similar results. Furthermore, it enlarges the class of signals that can be recovered without error from its M properly chosen samples to include those for which the Hadamard spectrum is either MBL or MBP components. Various bounds involving discrete and finite signals are estimated. These bounds are useful in signal processing and in approximating a given signal by a MBL signal.

References

1. Kak, S.C., "Sampling Theorem in Walsh-Fourier Analysis", Electronics Letter, vol.6, #14, pp.447-448 (July 1970).
2. Gibbs, J.E., "Discrete Complex Walsh Functions", 1970 Workshop on Application of Walsh Functions, Washington D.C., pp. 108-122.
3. Gibbs, J.E., "Sine Waves and Walsh Waves in Physics," Proc. Applications of Walsh Functions, pp.260-274, Washington D.C., (1970)
4. Moharir, P.S. & Sarma, K.R., Prasada, B., "Amplitude bounds and Quantization Scheme in Walsh-Fourier Domain", Proc. Walsh Function Symposium 1971, pp. 142-150, Washington DC., (1971).
5. LeDinh, C.T., Le, P., Coulet, R., "Sampling Expansions in Discrete and Finite WF Analysis", Research Report No.72 DTG-1, Dept. of Electrical Engineering, University of Sherbrooke, Sherbrooke, Que., Canada.
6. Papoulis, A., "Limits on Band-limited Signals", IEEE Proc., pp.1677-1686, (October 1967).

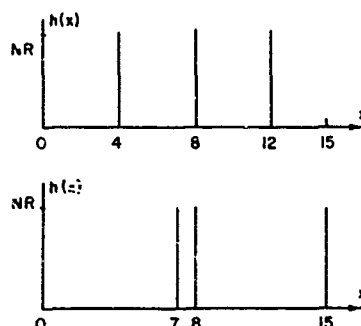


Fig.1 a-PERIODIC SAMPLING POINTS USING EQ 24
b-SAMPLING POINTS USING EQ. 26

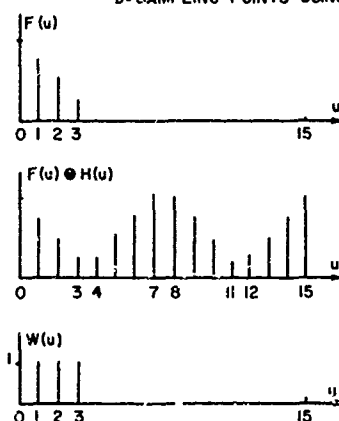


Fig.2 PROOF OF STATEMENT 4
a- ORIGINAL SPECTRUM
b- SPECTRUM OF SAMPLED SIGNAL
c- SPECTRUM OF FILTERING KERNEL

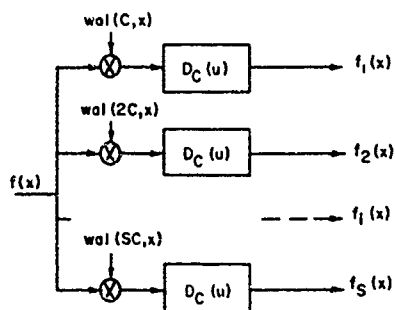


Fig. 3 GENERATING OF CBL ASSOCIATED SIGNALS



Fig. 4 a - A M-NON ZERO WALSH SPECTRUM
b - ITS HADAM/ RD MBL SPECTRUM

FAST "IN PLACE" COMPUTATION OF THE DISCRETE WALSH TRANSFORM IN SEQUENCY ORDER

Gunter Berauer

Institut für Elektrische Nachrichtentechnik
der Technischen Hochschule Aachen
Aachen, West Germany

1. Introduction

Walsh transforms have become a very important instrument in science and technique during the recent years. Especially in filter theory [1,8,9], picture processing [4] and pattern recognition [10] its application increased rapidly. SeQUENCY theory leads to a generalisation of filter concepts, and in signal processing the possibility of redundancy reduction by means of Walsh transforms is a main topic.

Caused by the high data rates of natural signals (as speech or image signals) computer aided methods must use fast and storage saving algorithms. Fast "in place" algorithms to perform the Walsh transform in unordered form are well known [5,6]. But especially in filter technique the order of seQUENCY related to the Walsh coefficients is important. Until now a fast "in place" algorithm to compute the Walsh coefficients in order of seQUENCY has not been published. In [11] Ulman even asserts, that such an in place algorithm cannot exist, and one has to spend at least $2 \cdot M$ storage locations for M input samples. On the contrary this paper describes a fast in-place algorithm to perform the ordered discrete Walsh transform using only a few more storage locations than input samples.

2. Discrete Walsh Transforms

Let \mathbb{Z}^1 be the set of complex numbers, and A be the set of discrete functions $x \in A$ with the following properties:

$$x(k) \in \mathbb{Z}^1 \text{ for all } k \in B_N := \{0, \dots, M-1\} \quad (1)$$

$$M = 2^N; N \geq 0; N \text{ integer}; N < \infty \quad (2)$$

$$\|x\| := \sum_{k=0}^{M-1} |x(k)|^2 < \infty \quad (3)$$

In A the set

$$E := \{y_u | u \in B_N\} \quad (4)$$

form a complete orthogonal set, if

$$y_u(k) = (-1)^{\langle k, u \rangle} \quad (5)$$

with

$$\langle k, u \rangle = \sum_{i=0}^{N-1} k_i u_i \quad (6)$$

$$k = (k_{N-1} k_{N-2} \dots k_0)_2 \quad (7)$$

and

$$u = (u_{N-1} u_{N-2} \dots u_0)_2 \quad (8)$$

are the binary representations of k and u respectively.

The transform

$$D_x(u) = \sum_{k=0}^{M-1} y_u(k) x(k) \in A \quad (9)$$

for all $u \in B_N$, and for each $x \in A$, may be called "unordered Walsh transform". The basis functions y_u are discrete Walsh functions.

Let L_N be a one-to-one function, $L_N : B_N \rightarrow B_N$, with the property, that for $\bar{u} := L_N(u)$ holds:

$$\bar{u}_i = \begin{cases} u_{N-1} & \text{for } i = 0 \\ u_{N-i} \oplus u_{N-i-1} & \text{for } i \in \{1, \dots, N-1\} \end{cases} \quad (10)$$

\oplus denotes mod 2 addition.

The Walsh coefficients $\bar{D}_x(u)$ of function x in order of seQUENCY are computed by means of

$$\begin{aligned} \bar{D}_x(u) &= D_x(\bar{u}) = D_x(L_N(u)) \\ &= \sum_{k=0}^{M-1} y_{\bar{u}}(k) x(k) \end{aligned} \quad (11)$$

The basis functions $y_{\bar{u}}$ can be written in the form

$$y_{\bar{u}}(k) = (-1)^{\langle \bar{u}, k \rangle} = \text{wal}_{\bar{u}}(k) \quad (12)$$

The eight discrete Walsh functions $\text{wal}_u(k)$ in seQUENCY order with respect to $M = 8$ are shown in figure 1. Equation (11) may be called Walsh transform in seQUENCY order.

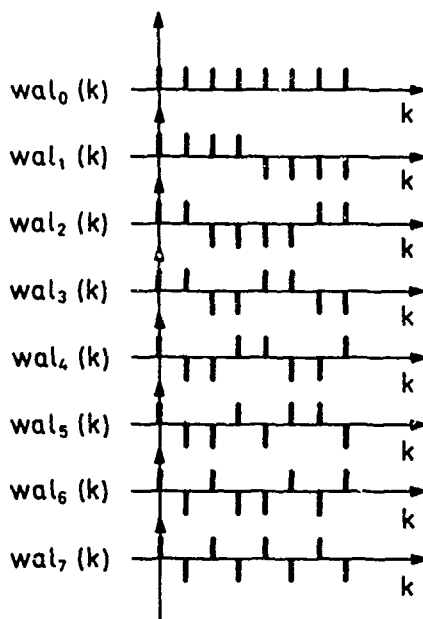


Fig. 1: Discrete Walsh Functions in Order of Sequency. $N = 3$, $M = 8$.

3. Special Properties of the Function

L_N

The following properties of L_N , which can be proved easily, are helpful in the later discussions.

A)
$$L_N(u) = BR_N(u \oplus SR(u)) \quad (13)$$

BR_N : bit reversal

SR : shift ight one bit (left zero filled)

\oplus : bit per bit mod 2 addition

$u \in B_N$

B) Inverse function L_N^{-1} (Def:

$L_N^{-1}(L_N(k)) = k$:

$$L_N^{-1}(u) = \sum_{r=0}^{N-1} SR^r[BR(u)] \quad (14)$$

\sum : bit per bit mod 2 summation

SR^r : shift right r bit (left zero filled)

$u \in B_N$

C) Let $\bar{u} = L_N(u)$ and $\bar{k} = L_N(k)$, then

$$y_{\bar{u}}(k) = y_u(\bar{k}) \quad (15)$$

$k, u \in B_N$

D) $L_N(0) = 0$ for all integer $N \geq 0$ (16)

E) There exists one and only one $0 \neq k_0(N) \in B_N$, with

$$L_N(k_0(N)) = k_0(N) \quad (17)$$

if and only if

$N = 3s+1$ for all integer $s > 0$, (18)

having the value

$$k_0(N) = 5 \cdot \sum_{r=0}^{s-1} 2^{3r} + 2^{N-1} \quad (19)$$

The indices $k_0(N)$ and 0 (zero) shall be called "fix points" with respect to L_N .

4. Fast Algorithms and Assorting Problem

Fast algorithms to compute the unordered Walsh coefficients "in place" are well known [5,6]. Having computed the coefficients in unordered form, they can be assorted into order of sequency (decimation in sequency). On the other hand it is possible to assort the input values $x(k)$, $k \in B_N$ in such a way, that the unordered transform applied to this shuffled input sequence yields coefficients in order of sequency (decimation in time).

The assortment is simply done by use of a second memory field (\bar{D}_x or \bar{x}):

A) Decimation in Sequency

Following equation (11) the assorting algorithm is:

$$\forall u \in B_N : D_x(u) \rightarrow \bar{D}_x(L_N^{-1}(u)) \quad (20)$$

B) Decimation in Time

Inserting equation (15) into (11) yields

$$\bar{D}_x(u) = \sum_{k=0}^{M-1} y_u(L_N(k))x(k) \quad (21)$$

or

$$\begin{aligned} \bar{D}_x(u) &= \sum_{k1=0}^{M-1} y_u(k1)x(L_N^{-1}(k1)) \\ &= \sum_{k=0}^{M-1} y_u(k)\bar{x}(k) = D_{\bar{x}}(u) \end{aligned} \quad (22)$$

Hence the assorting algorithm is

$$\forall k \in B_N : x(k) \rightarrow \bar{x}(L_N(k)) \quad (23)$$

5. "In Place" Solution of the Assorting Problem.

Assorting "in place" is a problem, because in general

$$L_N^2(k) := L_N(L_N(k)) \neq k \quad (24)$$

That's why assorting can't be done only by multiple exchanging of two input samples or coefficients (like "bit reversal" assorting in FFT algorithms).

But with respect to the function L_N , defined in (10) or (13) a "subsequent" "in place" assorting using two further memory locations (H1 and H2) is possible.

Example:

Let $N = 3$, $M = 8$ then:

$$L_3(1)=4, L_3(4)=3, L_3(3)=2, L_3(2)=6, \\ L_3(6)=5, L_3(5)=7, L_3(7)=1$$

or:

$$L_3^1(1)=4, L_3^2(1)=3, \dots, L_3^7(1)=1$$

Figure 2 shows the necessary cyclic shuffling for decimation in time.

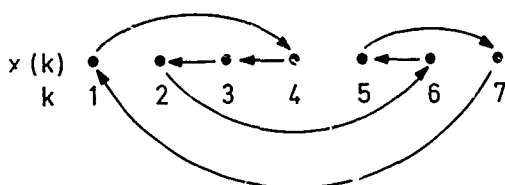


Fig. 2: Assorting "In Place" in the Case $N = 3$

Starting the "in place" assorting at index 1, decimation in time is performed by the following steps ($\langle H1 \rangle$ denotes "content of memory location with symbolic address H1"):

$$\begin{array}{ll} \text{Start: } x(1) \rightarrow H2 & \\ x(4) \rightarrow H1 & \\ \langle H2 \rangle \rightarrow 4 & \\ \langle H1 \rangle \rightarrow H2 & \left. \begin{array}{l} \\ \\ \end{array} \right\} 1 \\ \vdots & \vdots \\ x(L_3^k(1)) \rightarrow H1 & \\ H2 \rightarrow L_3^k(1) & \left. \begin{array}{l} \\ \\ \end{array} \right\} k \\ H1 \rightarrow H2 & \\ \vdots & \vdots \end{array}$$

$$\begin{array}{ll} \vdots & \\ x(7) \rightarrow H1 & \\ \langle H2 \rangle \rightarrow 7 & \\ \langle H1 \rangle \rightarrow H2 & \left. \begin{array}{l} \\ \\ \end{array} \right\} 6 \\ \vdots & \end{array}$$

Stop: $\langle H2 \rangle \rightarrow 1$

In the example $N=3$ the whole input field x is assorted in one "cycle" beginning and ending at an available $k_1 \in B_3 - \{0\}$ (here: $k_1=1$).

For $N = 4$ there exist two cycles:

$$F_1^{(4)} = \{1, 8, 3, 4, 6, 10, 15\} \\ F_2^{(4)} = \{2, 12, 5, 14, 9, 11, 7\}$$

The total input field is assorted by running through both cycles starting at available entry points $k_1 \in F_1^{(4)}$ and $k_2 \in F_2^{(4)}$. Because 0 and 13 are fix points, $x(0)$ and $x(13)$ remain unchanged.

Introducing fix points (index 0 and perhaps one more fix point $k_0(N)$) to be cycles of length 1, these cycles form a partition of B_N defined by the function L_N .

A general formulation of this fact is as follows:

For each $N \geq 1$, integer, and $M = 2^N$ exists a set $C^{(N)} = \{1, 2, \dots, r(N)\}$ and a partition $\{F_i^{(N)} | i \in C^{(N)}\}$ of B_N with the property:

$$\forall i \in C^{(N)} \quad \exists u(i) \in \{1, \dots, M-1\} \quad \forall k \in F_i^{(N)} : \\ a) L_N^{u(i)}(k) = k \quad \text{and} \quad (25)$$

$$b) \quad \forall u(i) \neq 1 \quad \forall s \in \{1, \dots, u(i)-1\}$$

$$: L_N^s(k) \neq k \quad (26)$$

Now let $G^{(N)} = \{k_i | i \in C^{(N)}\}$; $k_i \in F_i^{(N)}$ be a set of entry points belonging to the cycles $F_i^{(N)}$. Decimation in time is then performed by

$$\forall k_i \in G^{(N)} : x(k_i) \xrightarrow{H1, H2} x(L_N^1(k_i)) \xrightarrow{H1, H2} \dots \\ \xrightarrow{H1, H2} x(L_N^{u(i)}(k_i)) = x(k_i) \quad (27)$$

and decimation in sequency by

$$\forall k_i \in G(N) : D_x(k_i) \xrightarrow{H1, H2} D_x(L_N^{-1}(k_i)) \xrightarrow{H1, H2} \dots \xrightarrow{H1, H2} D_x(L_N^{-u(i)}(k_i)) = D_x(k_i) \quad (28)$$

6. Realization in a Computer Program

Until now we didn't find an explicit expression to compute a set of entry points of the cycles for arbitrary choice of N . Therefore entry points for all $1 \leq N \leq 14$ were computed in a separate search program.

For $N = 1, 2, 3, 5, 6, 9, 11, 14$ the number of cycles is $r(N) = 2$, i.g. $C(N) = \{2\}$, and $F_1^{(N)} = B_N - \{0\}$, $F_2^{(N)} = \{0\}$. Because 0 is a fix point only $F_1^{(N)}$ is of interest. An arbitrary entry point $k_i \in F_1^{(N)} = B_N - \{0\}$ can be chosen (we chose $k_1 = 1$).

For $N = 4, 7, 8, 10, 12, 13$ is $r(N) > 2$. Table 1 shows the computed entry points (except the fix point zero).

N :	4	7	8	10	12	13
1	1	1	1	1	1	1
2	2	2	2	2	4	2
13	4	4	7	19	7	7
(fix)	7	7	8	26	11	11
	11	8	11	1585	13	13
	13	11	13		16	16
	14	13	16		19	19
	19	14	19		22	22
	22	19	22		25	25
	49	21	26		26	26
	109	22	28		35	35
(fix)	26	35	38		38	38
	28	38	41		41	41
	35	41	69		69	69
	49	55	81		81	81
	52	91	100		100	100
	59	110	769		769	769
	81	193	1154		1154	1154
	117	877	7021		7021	7021
	(fix)	(fix)	(fix)		(fix)	(fix)

Table 1: Entry Points in the Case of Multiple Cycles with $r(N) > 2$.

Excluding fix points and the number 1, being entry point of the first cycle for all N , the entry points of table 1 are stored in the fast "in place" Walsh subroutine by a data declaration. The subroutine uses decimation in time. On a CDC 6400 computer the total subroutine length is 27210 memory locations (excluding the input

data field).

We also wrote a Walsh subroutine using a second field into which the input data is shuffled (no "in place" method). This subroutine has a total length of 14910 memory locations (excluding the two data fields).

Therefore the "in place" algorithm is memory saving for all $N \geq 7$.

The computation times of both subroutines are nearly the same.

References.

- [1] Harmuth, H.F., "Sequenzfilter für Signale mit zwei Raumvariablen und LCS-Filter", Nachrichtentechn. Z. 23 (1970), No. 8, pp. 377-383.
- [2] Ahmed, N., Bates, R.M., Rao, K.R., "Multidimensional Bifore Transform", Electronic Letters Vol. 6, No. 8, 16th April 1970, pp. 237-238.
- [3] Ahmed, N., Rao, K.R., "Discrete Fourier and Hadamard Transforms", Electronic letters Vol. 6, No. 7, 2nd April 1970, pp. 221-224.
- [4] Pratt, W.K., Kane, J., Andrews, H.C., "Hadamard Transform Image Coding". Proc. IEEE Vol. 57 (1969), No. 1, pp. 58-68.
- [5] Shanks, J.L., "Computation of the fast Walsh-Fourier-Transform", IEEE Trans.Comp. May 1969, pp. 457-459.
- [6] Georgi, K.H., "Ein Schema für die schnelle Walsh-Transformation", Nachrichtentechn. Z. 24 (1971), No. 9, pp. 461-463.
- [7] Ahmed, N., Rao, K.R., Abdussattar, A.L., "Bifore or Hadamard Transform", IEEE AU-19(1971), No.3, pp. 225-234.
- [8] Berauer, G., "Eine Methode zur Analyse und Synthese diskreter Walsh-Filter", Nachrichtentechn. Z. 24 (1971), No. 11, pp. 569-571.
- [9] Berauer, G., "Eine allgemeine Theorie diskreter Filter", AEU 25 (1971), pp.575-578.
- [10] Andrews, H.C., "Some Unitary Transformations in Pattern Recognition and Image Processing", Proc. IFIP Congress 1971, Ljubljana, Booklet TA-2, pp. 70-74.
- [11] Ulman, L.J., "Computation of the Hadamard Transform and the R-Transform in Ordered Form", IEEE Trans.Comp. April 1970, pp. 359-360.

EFFICIENT COMPUTATION OF THE WALSH-

HADAMARD TRANSFORM SPECTRAL MODES

N. Ahmed
Depts. of Elec. Engrg.
& Computer Science
Kansas State Univ.
Manhattan, Kansas

A. L. Abdussattar
Dept. of Elec. Engrg.
Kansas State Univ.
Manhattan, Kansas

K. R. Rao
Dept. of Elec. Engrg.
Univ. of Texas at Arlington
Arlington, Texas

INTRODUCTION

The Walsh-Hadamard or BIFORE (Binary Fourier Representation) transform and related spectra [1] have found several applications in pattern recognition problems [2, 3]. Recently Ohnsorg [4] defined additional spectral modes for the Walsh-Hadamard transform (WHT) which could also find applications in pattern recognition problems. These spectral modes are (i) the quadratic spectrum and (ii) the optimum quadratic spectrum. However, it is a laborious task to compute these spectra. To this end, the main objective of this paper is to present efficient algorithms which enable rapid computation of the WHT quadratic and optimum quadratic spectra.

WALSH-HADAMARD TRANSFORM (WHT)

Consider a real-valued N -periodic data sequence $\{x(n)\} = \{x_0, x_1, \dots, x_{N-1}\}$, $n = 0, 1, \dots, N-1$. If the vector representation of $\{x(n)\}$ is denoted by $\{x(n)\}$, then

$$\{x(n)\}' = \{x_0 \ x_1 \ \dots \ x_{N-1}\}$$

where $\{x(n)\}'$ denotes the transpose of $\{x(n)\}$. The WHT of $\{x(n)\}$ is defined as [1]

$$\{X(n)\} = \frac{1}{N} [H(n)] \{x(n)\} \quad (1)$$

where $\{X(n)\}' = \{X_0 \ X_1 \ \dots \ X_{N-1}\}$ is the WHT vector, and $[H(n)]$ is the $(N \times N)$ Hadamard matrix. Let $\{x^{(l)}(n)\}$ denote the sequence obtained by shifting $\{x(n)\}$ by l positions to the left. That is,

$$\{x^{(l)}(n)\}' = \{x_l \ x_{l+1} \ \dots \ x_{N-1}\}, \quad l = 1, 2, \dots, (N-1) \quad (2)$$

If $\{X^{(l)}(n)\}' = \{X_0^{(l)} \ X_1^{(l)} \ \dots \ X_{N-1}^{(l)}\}$ denotes the WHT vector of $\{x^{(l)}(n)\}$, then it can be shown that [1]

$$\{X^{(l)}(n)\} = [A(n)]^l \{X(n)\} \quad (3)$$

where $[A(n)] = \frac{1}{N} [H(n)] [M(n)] [H(n)]$,

$[A(n)]^l$ is $[A(n)]$ raised to the power l , and

$[M(n)]$ is the $(2^n \times 2^n)$ identity matrix, whose columns are shifted cyclically to the right by one place.

The matrix $[A(n)]$ is called the WHT "shift matrix" since it relates the WHT of the shifted sequence to that of the original sequence. It has a "block diagonal" structure since it is of the form

$$[A(n)] = [\bar{B}(0)] \oplus [B(0)] \oplus [B(1)] \oplus \dots \oplus [B(n-1)]_{(4)}$$

where, $[\bar{B}(0)] = [B(0)] = 1$, and each of the $(2^m \times 2^m)$ matrices $[B(m)]$, $m = 1, 2, \dots, (n-1)$ is orthogonal.

SPECTRAL MODES OF THE WHT

The shift matrix $[A(n)]$ in (3) yields the familiar WHT power spectrum in a straightforward manner [1]. However, two additional modes, namely the WHT quadratic and optimum quadratic spectra [5], have also been developed.

(i) The WHT quadratic spectrum: This spectrum is defined as

$$Q_0 = X_0^2$$

$$Q_{m,q} = \{X_k\}_m' [B(m-1)]^q \{X_k\}_m \quad (5)$$

$$2^{m-1} \leq k < 2^m, \quad 0 \leq q < 2^{m-2}, \quad 0 < m \leq n$$

where

$$\{X_k\}_m = \{X_{2^{m-1}} \ X_{2^{m-1}+1} \ \dots \ X_{2^m-1}\}$$

In what follows, the WHT quadratic spectrum will be referred to as the Q -spectrum. It can be shown that [5] the Q -spectrum is invariant to cyclic shift of the data sequence $\{x(n)\}$.

For example, consider the case $N=8$. Then, (5) yields

$$Q_0 = X_0^2$$

$$Q_{1,0} = X_1^2$$

$$Q_{2,0} = X_2^2 + X_3^2$$

$$Q_{3,0} = X_4^2 + X_5^2 + X_6^2 + X_7^2$$

$$Q_{3,1} = \frac{1}{2} \{X_4^2 - X_5^2 + X_6^2 - X_7^2 - 2X_4X_7 + 2X_5X_6\} \quad (6)$$

From (6) it follows that in general, (5) yields a set of $(\frac{N}{2} + 1)$ shift invariants each of which is a quadratic form. Thus, as N increases, the storage requirements and computational time increases rapidly. Consequently alternate methods which alleviate both these problems are desirable. In this paper, three such methods are presented and their performance is compared.

(ii) The WHT optimum quadratic spectrum: Consider the case $N=8$. Then the WHT optimum quadratic spectrum corresponding to (6) is given by [5],

$$J_0 = Q_0$$

$$J_{1,0} = Q_{1,0}$$

$$J_{2,0} = Q_{2,0}$$

$$\begin{bmatrix} J_{3,0} \\ J_{3,1} \end{bmatrix} = [T(1)]^{-1} \begin{bmatrix} Q_{3,0} \\ Q_{3,1} \end{bmatrix}$$

where, $[T(1)] = \begin{bmatrix} 1 & 1 \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix}$

or,

$$J_{3,0} = x_4^2 + x_6^2 - x_4 x_7 + x_5 x_6$$

$$J_{3,1} = x_5^2 + x_7^2 + x_4 x_7 - x_5 x_6. \quad (7)$$

From (7) it follows that the squared terms in $J_{3,0}$ and $J_{3,1}$ represent the individual sequences 1 and 3 respectively. In contrast, $Q_{3,0}$ and $Q_{3,1}$ represent the group of sequences 1 and 3. In general it can be shown that each $J_{m,q}$ represents a single sequence in its squared terms besides being shift invariant. Hence the name "Optimum" quadratic spectrum. For convenience, this spectrum will be referred to as the J-spectrum.

From (7) it is clear, that in general, the computation of the J-spectrum is tedious, since it involves both the development and subsequent inversion of the transformation matrix $[T(m)]$ whose order increases with N . The general procedure to develop $[T(m)]$, which is presented in [5], requires the computation of powers of the submatrices $[B(m)]$ in (4), which is a laborious task. However, in this paper it is shown that the J-spectrum can be rapidly computed using an alternate approach.

In what follows, methods to compute the Q-spectrum and the J-spectrum are presented and compared. In order to be brief, details pertaining to the derivations of the related algorithms are avoided.

COMPUTATION OF THE Q-SPECTRUM

Three methods can be used to compute the Q-spectrum.

Method I: From (3) and (4) it follows that (5) can be written as

$$Q_0 = x_n^2$$

$$Q_{m,q} = \{x_k\}_m' \{x_k^{(q)}\}_m \quad (8)$$

$$2^{m-1} \leq k < 2^m, \quad 0 \leq q < 2^{m-2}, \quad 0 < m \leq n$$

where

$$\{x_k^{(q)}\}_m = \{x_{2^{m-1}-1}^{(q)} \quad x_{2^{m-1}+1}^{(q)} \dots x_{2^m-1}^{(q)}\}.$$

The Q-spectrum as defined in (8) do not require that the matrices $[B(m)]$ be stored, as is the case in (5). Again, the $\{x_k^{(q)}\}$ can be computed by successively applying the fast Walsh-Hadamard transform (FWHT) q times to $\{x(n)\}$ shifted cyclically to the left, q times.

Method II: Consider the modified BT or modified WHT which is defined as [6]

$$\{F(n)\} = \frac{1}{N} [D(n)] \{x(n)\}, \quad (9)$$

where $\{x(n)\}$ is the data sequence, and $\{F(n)\}$ is the transform vector such that,

$$\{F(n)\}' = \{F_0 \quad F_1 \dots F_{N-1}\},$$

and $[D(n)]$ is defined by recurrence relation

$$[D(k+1)] = \begin{bmatrix} [D(k)] & -[D(k)] \\ 2^{k/2}[I(k)] & -2^{k/2}[I'(k)] \end{bmatrix},$$

$$k = 0, 1, \dots, n$$

with $[D(0)] = 1$

The transform coefficients F_k , $k = 0, 1, \dots, (N-1)$

in (9) can be computed rapidly as shown in Fig. 1 for the case $N=8$.

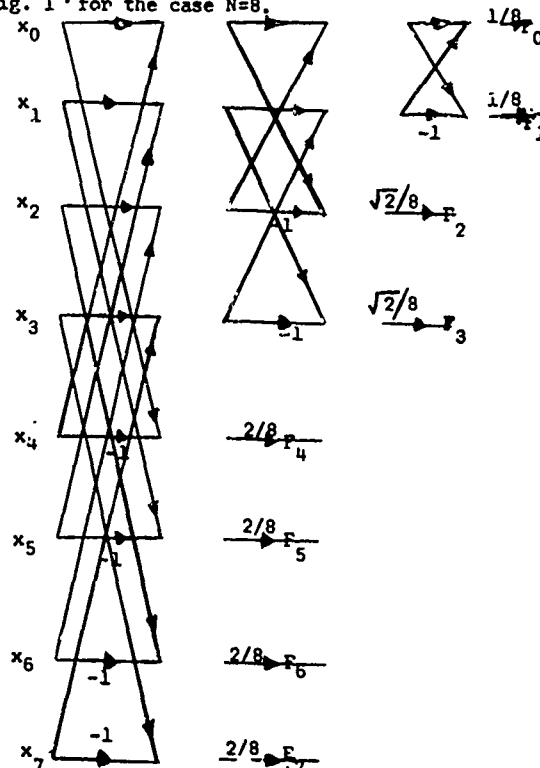


Fig. 1. Signal flow graph for the modified WHT for $N=8$.

It can be shown that the Q-spectrum can be expressed in terms of F_k , $k = 0, 1, \dots, (N-1)$ as follows:

$$Q_0 = F_0^2$$

$$Q_{m,0} = \sum_{k=2^{m-1}}^{2^m-1} F_k^2, \quad 0 < m \leq n$$

$$Q_{m,q} = \sum_{k=2^{m-1}}^{2^m-1} F_{k+q}^2 - \sum_{j=0}^{q-1} F_{2^{m-1}+j}^2 F_{2^m+j-q}^2 \quad (10)$$

$$2^{m-1} \leq k < 2^m, \quad 0 < q \leq 2^{m-2}, \quad 0 < m \leq n.$$

The Q-spectrum as defined in (10) can be computed extremely rapidly since it involves successive cyclic shifts, sign changes and dot products which is best illustrated by an example. Consider $N=16$. Then (10) yields,

$$Q_0 = F_0^2$$

$$Q_{1,0} = F_1^2$$

$$\begin{aligned}
Q_{2,0} &= F_2^2 + F_3^2 \\
Q_{3,0} &= F_4^2 + F_5^2 + F_6^2 + F_7^2 \\
Q_{3,1} &= \begin{pmatrix} F_4 & F_5 & F_6 & F_7 \\ F_5 & F_6 & F_7 & F_4 \end{pmatrix} = F_4 F_5 + F_5 F_6 + F_6 F_7 \\
Q_{4,0} &= F_8^2 + F_9^2 + \dots + F_{15}^2 \\
Q_{4,1} &= \begin{pmatrix} F_8 & F_9 & \dots & F_{14} & F_{15} \\ F_9 & F_{10} & \dots & F_{15} & F_8 \end{pmatrix} \\
&= \sum_{k=8}^{14} F_k F_{k+1} - F_8 F_{15} \\
Q_{4,2} &= \begin{pmatrix} F_8 & F_9 & \dots & F_{14} & F_{15} \\ F_{10} & F_{11} & \dots & F_8 & F_9 \end{pmatrix} \\
&= \sum_{k=8}^{13} F_k F_{k+2} - (F_8 F_{14} + F_9 F_{15}) \\
Q_{4,3} &= \begin{pmatrix} F_8 & F_9 & \dots & F_{13} & F_{14} & F_{15} \\ F_9 & F_{10} & \dots & F_8 & F_9 & F_{10} \end{pmatrix} \\
&= \sum_{k=8}^{12} F_k F_{k+3} - (F_8 F_{13} + F_9 F_{14} + F_{10} F_{15})
\end{aligned}$$

Method III: Consider the following modification of the Q-spectrum as defined in (5):

$$\begin{aligned}
Q_0 &= X_0^2 \\
Q_{m,q} &= \{X_k\}_m' [B(m-1)]^q \{X_k\}_m \quad (12) \\
2^{m-1} \leq k < 2^m, \quad 0 \leq q < 2^{m-1}, \quad 0 < m \leq n.
\end{aligned}$$

For example when $N=8$, (12) yields,

$$\begin{aligned}
Q_0 &= X_0^2 \\
Q_{1,0} &= X_1^2 \\
Q_{2,0} &= X_2^2 + X_3^2 \\
Q_{2,1} &= 0 \\
Q_{2,2} &= -Q_{2,0} \\
Q_{3,0} &= X_4^2 + X_5^2 + X_6^2 + X_7^2 \\
Q_{3,1} &= \frac{1}{2} \{X_4^2 - X_5^2 + X_6^2 - X_7^2 - 2X_4 X_7 \\
&\quad + 2X_5 X_6\} \\
Q_{3,2} &= 0 \\
Q_{3,3} &= -Q_{3,1} \quad (13)
\end{aligned}$$

Inspection of (13) shows that $N=8$ yields $5 = \binom{8}{2} + 1$ independent Q-spectrum points namely $Q_0, Q_{1,0}, Q_{2,0}, Q_{3,0}$ and $Q_{3,1}$ which agree with those in (6). In general, (12) yields $\binom{N}{2} + 1$ independent Q-spectrum points which are identical to those defined by (5).

The definition of the Q-spectrum in (12) yields the following WHT autocorrelation theorem:

$$\{\hat{Q}(n)\} = \frac{1}{N} [D(n)] \{r(n)\} \quad (14)$$

where

$$\begin{aligned}
\{\hat{Q}(n)\}' &= \{\hat{Q}_0, \hat{Q}_{1,0}, \hat{Q}_{2,0}, \hat{Q}_{2,1}, \dots, \hat{Q}_{n,2^{n-1}-1}\}, \\
\text{with } \hat{Q}_0 &= Q_0, \quad \hat{Q}_{1,0} = Q_{1,0} \\
\hat{Q}_{m,l} &= 2^{-(m-1)/2} Q_{m,l} \quad \text{for } 2 \leq m \leq n, \\
&\quad 0 \leq l < 2^{m-1}-1
\end{aligned}$$

and

$\{r(n)\}' = \{r_0, r_1, \dots, r_{N-1}\}$ is the autocorrelation vector whose elements are the cyclic autocorrelation coefficients of the data sequence $\{x(n)\}$. The transformation matrix $[D(n)]$ is defined in (9).

From (14) it is clear that the Q-spectrum can be rapidly computed as follows:

1. Compute $r_k, k=0,1,\dots,(N-1)$ using the fast Fourier transform (FFT) technique.
2. Compute $\hat{Q}_{m,l}$ as in (14) using the modified WHT defined in (9).

SUMMARY OF COMPUTATIONAL RESULTS FOR THE Q-SPECTRUM

Table I represents a summary of execution times to compute the Q-spectrum using Methods I-III discussed above. An IBM 360/50 digital computer was used.

N	Execution time in minutes		
	Method I	Method II	Method III
32	< 0.12	< 0.12	< 0.12
64	< 0.12	< 0.12	< 0.12
128	0.18	< 0.12	< 0.12
256	0.42	< 0.12	0.18
512	1.80	0.24	0.36
1024	7.62	0.54	0.78

Table 1. Comparison of the execution times for computation of the Q-spectra. From Table I it follows that Method II is the most efficient with respect to execution time for the values of N considered. Memory requirements for the three methods are approximately the same. Also, Method II which uses the modified WHT has an additional advantage over Method III in that it requires only real number arithmetic.

COMPUTATION OF THE J-SPECTRUM

From (7) it follows that the J-spectrum is obtained from the Q-spectrum by means of a linear transformation. The general form of the transform is as follows [5]:

$$\begin{aligned}
J_0 &= Q_0 \\
J_{1,0} &= Q_{1,0} \\
J_{2,0} &= Q_{2,0} \\
\{J_s\}_q &= [T(s-2)]^{-1} \{Q_s\}_q \quad (15)
\end{aligned}$$

$$C \leq q < 2^{s-2}, \quad 3 \leq s \leq n$$

where,

$$\{J_s\}_q = \{J_{s,0}, J_{s,1}, \dots, J_{s,2^{s-2}-1}\}$$

$$\{Q_s\}_q = \{Q_{s,0}, Q_{s,1}, \dots, Q_{s,2^{s-2}-1}\}$$

and $[T(n-2)]$ is a $(2^{s-2} \times 2^{s-2})$ matrix whose

q-th row is equal to the upper half diagonal of the $[B(s-1)]^q$ where $[B(s-1)]$ is defined in (4).

Let $[I(n)]$ denote a shifted version of $[I(n)]$ with the all zeros in the first column, such that

$$[I(n)] = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \\ & & \ddots & & \\ 0 & 1 & 0 & \cdots & 0 \end{bmatrix} \quad (16)$$

For example, for $N=8$, (16) yields

$$[I(2)] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Then, it can be shown that the $\{Q_s\}$ to $\{J_s\}$ transformation can be expressed as follows:

$$\begin{aligned} J_0 &= Q_0 \\ J_1 &= Q_{1,0} \\ J_{2,0} &= Q_{2,0} \\ \{J_s\}_q &= \frac{1}{2^{s-2}} [W(s-2)] \{Q_s\}_q, \quad 0 \leq q < 2^{s-2}, \\ &\quad 3 \leq s \leq n \end{aligned}$$

where, $[W(s-2)]$ has the recurrence relation,

$$[W(k+1)] = \begin{bmatrix} W(k) & | & 2W(k) - W(k) \hat{I}(k) \\ \hline W(k) & | & -2W(k) - W(k) \hat{I}(k) \end{bmatrix},$$

$$k = 0, 1, \dots, (n-3). \quad (17)$$

The recurrence relation in (17) provides the key to an efficient algorithm. For example, consider $N=32$. Then the recurrence relation is used thrice with $s=3, 4$ and 5 . As an illustration, the sequence of computations for $s=5$ are shown in Fig. 2. Since $[W(s-2)]$ is a (8×8) matrix when $s=5$, there are 3 iterations. In general, for a given s , $3 \leq s \leq n$ there are $\log_2(s-2)$ iterations. From Fig. 2 it follows that in general, the number of arithmetic operations and storage locations required is proportional to N , as in the case with the FWHT and the FFT. In this sense, the above J-spectrum algorithm is optimum. The execution times associated with this algorithm for various values of N are summarized in Table II. A 360/50 IEM digital computer was used.

N	32	64	128	256	512	1024
Execution time in minutes	<0.18	0.18	0.24	0.36	0.6	1.62

Table 2. Execution time for computation of the J-spectrum.

CONCLUSIONS

The methods presented in this paper for computing the WHT quadratic and optimum quadratic spectra can be used for the purposes of conducting

economical feasibility studies in pattern recognition problems. The WHT or BIFORE power spectrum [1] which is also a WHT spectral mode has already found several applications in the general area of pattern recognition [2-4].

REFERENCES

- [1] N. Ahmed, K.R. Rao and A.L. Abdussattar, "BIFORE or Hadamard transform," IEEE Trans. Audio and Electroacoustics, vol. AU-19, Sept. 1971. pp. 225-234.
- [2] R. M. Larson, "Feasibility study of an Automatic Signal Classification Technique," Honeywell document No. D-6D-6-15, MR 9535, March 2, 1966.
- [3] R. M. Powell: "Feasibility Study of an Active Sonar Signal Classification Technique," Honeywell document No. D-6D-G-16, MR 9519, March 11, 1966.
- [4] P. Milne, et al, "Application of a Frequency Approach to the Classification of Electrocardiograph Signals," To be presented at the 1972 Symposium on the Applications of Walsh Functions. Washington, D.C. March 27-29, 1972.
- [5] F. R. Ohnsorg, "Spectral Modes of the Walsh-Hadamard Transform," IEEE Trans. Electromagnetic compatibility, vol. EMC-13, August 1971. pp. 55-59.
- [6] N. Ahmed, and R. B. Schultz, "Position Spectrum Considerations," IEEE Trans. Audio and Electroacoustics, vol. Au-19, December 1971, pp. 326-327.

Iteration 1	Iteration 2	Iteration 3
$Q_{5,0} = Y(0) \rightarrow Y(0) + 2Y(4) - 0 = Y_1(0) \rightarrow Y_1(0) + 2Y_1(2) - 0 = Y_2(0) \rightarrow Y_2(0) + Y_2(1) = Y_3(0) \xrightarrow{1/8} J_{5,0}$		
$Q_{5,1} = Y(1) \rightarrow Y(1) + 2Y(5) - Y(7) = Y_1(1) \rightarrow Y_1(1) + 2Y_1(3) - Y_1(3) = Y_2(1) \rightarrow Y_2(0) - Y_2(1) = Y_3(1) \xrightarrow{1/8} J_{5,1}$		
$Q_{5,2} = Y(2) \rightarrow Y(2) + 2Y(6) - Y(6) = Y_1(2) \rightarrow Y_1(2) - 2Y_1(2) - 0 = Y_2(2) \rightarrow Y_2(2) + Y_2(3) = Y_3(2) \xrightarrow{1/8} J_{5,2}$		
$Q_{5,3} = Y(3) \rightarrow Y(3) + 2Y(7) - Y(4) = Y_1(3) \rightarrow Y_1(3) - 2Y_1(3) - Y_1(3) = Y_2(3) \rightarrow Y_2(2) - Y_2(3) = Y_3(3) \xrightarrow{1/8} J_{5,3}$		
$Q_{5,4} = Y(4) \rightarrow Y(0) - 2Y(4) - 0 = Y_1(4) \rightarrow Y_1(4) + 2Y_1(6) - 0 = Y_2(4) \rightarrow Y_2(4) + Y_2(5) = Y_3(4) \xrightarrow{1/8} J_{5,4}$		
$Q_{5,5} = Y(5) \rightarrow Y(1) - 2Y(5) - Y(7) = Y_1(5) \rightarrow Y_1(5) + 2Y_1(7) - Y_1(7) = Y_2(5) \rightarrow Y_2(4) - Y_2(6) = Y_3(5) \xrightarrow{1/8} J_{5,5}$		
$Q_{5,6} = Y(6) \rightarrow Y(2) - 2Y(6) - Y(6) = Y_1(6) \rightarrow Y_1(4) - 2Y_1(6) - 0 = Y_2(6) \rightarrow Y_2(6) + Y_2(7) = Y_3(6) \xrightarrow{1/8} J_{5,6}$		
$Q_{5,7} = Y(7) \rightarrow Y(3) - 2Y(7) - Y(5) = Y_1(7) \rightarrow Y_1(5) - 2Y_1(7) - Y_1(7) = Y_2(7) \rightarrow Y_2(6) - Y_2(7) = Y_3(7) \xrightarrow{1/8} J_{5,7}$		

Fig. 2 Signal flowgraph associated with (17) for $s=5$.

ANALYSIS OF THE LOGICAL WALSH TRANSFORM FOR $N = 16$ *

William A. Parkyn, Jr.
Gordon E. Cash
Mathematical Sciences Department
McDonnell Douglas Astronautics Company
Huntington Beach, California

Abstract

Searle's Logical Walsh Transform, presented in 1970, is investigated in detail for sequences of longer length than the 4-bit sequences ($N = 4$) considered by Searle. This transform is attractive for processing binary data because, unlike the conventional Walsh transform, a logical transform contains no more bits than the original signal. The original goal of the investigation was the discovery of a fast Boolean algorithm for this transform which would require little intermediate storage. In searching for properties of the transform on which such an algorithm might be based, however, it was found that very simple properties and relationships which hold for $N = 4$ become messy and complex for sequences of longer length. In fact, transform reversibility even breaks down for $N \geq 16$. A detailed analysis of the behavior and reasons for the breakdown is presented, including a simple rule for determining the "singular points" of the transform.

Introduction

Searle (1970) introduced a "logical" Walsh transform, which eliminates redundancy resulting from the application of the conventional Walsh transform to binary data. In the general case, if the input data to the conventional Walsh transform have a dynamic range D , the transform coefficients that are the output data have a dynamic range of ND , where N is the "length" of the transform (i.e., the number of data points input to the transform). Original data that are quantized to m bits per data point will require mN bits per data point for complete representation in the transform domain.

Searle noted that if a binary data string ($m = 1$) of length $N = 4$ were input to the Walsh transform, and if the first bit is 1, only the sign bits of the transform output are necessary to completely restore the original binary data. Furthermore, it turned out that these sign bits need only be input to the Walsh transform and the sign bits of the output data were identical to the original binary data. The set of ordered sign bits is called the logical Walsh transform. The integer x corresponds to the set of N input data $\{x_1, x_2, \dots, x_N\}$ according to

$$x = \sum_{i=1}^N x_i 2^{N-i}$$

while the integer X corresponds to the bits of the logical transform $\{X_1, X_2, \dots, X_N\}$

$$X = \sum_{k=1}^N X_k 2^{N-k}$$

These bits $\{X_k\}$ are defined by the operation of thresholding:

$$\begin{aligned} X_k &= 1 \text{ if } \sum_{i=1}^N x_i \text{ WAL}(k,i) = W_k(x) > 0 \\ X_k &= 0 \text{ if } \sum_{i=1}^N x_i \text{ WAL}(k,i) = W_k(x) \leq 0 \end{aligned} \quad (1)$$

where $\text{WAL}(k,i)$ is the k th Walsh function in the Kronecker or periodicity (Parkyn, 1970) ordering, that of the fast Walsh transform algorithm. The convention that 0 in the Walsh transform is associated with the negative rather than the positive integers may at first appear arbitrary. Under the convention that the first bit of the sequence is 1, this association is necessary for transform invertibility. In fact, the transform is its own inverse

$$\text{LWAL}[\text{LWAL}(x)] = \text{LWAL}(X) = x.$$

Searle suggested that the transform be extended to binary sequences starting with a 0 bit by a complementation procedure. For $0 \leq x \leq 2^{N-1} - 1$ define

$$\text{LWAL}(x) = \hat{x},$$

where $\hat{x} = \text{LWAL}(x)$. The caret (^) is used to denote the complementation mapping $x \rightarrow (2^{N-1} - x)$.

In the rest of this paper, hexadecimal (base 16) numerals will often be used for convenience. Table 1 gives these numerals, their binary representation, the 4-bit logical transform, and the modulo-2 sum of each number and its logical transform.

*This work was sponsored by the Air Force Office of Scientific Research (Air Force Systems Command) under Contract F44620-71-C-0052.

Table 1

x	x ₁	x ₂	x ₃	x ₄	X	x⊗X
0	0	0	0	0	7	7
1	0	0	0	1	1	0
2	0	0	1	0	4	6
3	0	0	1	1	5	6
4	0	1	0	0	2	6
5	0	1	0	1	3	6
6	0	1	1	0	6	0
7	0	1	1	1	0	7
8	1	0	0	0	F	7
9	1	0	0	1	9	0
A	1	0	1	0	C	6
B	1	0	1	1	D	6
C	1	1	0	0	A	6
D	1	1	0	1	B	6
E	1	1	1	0	E	0
F	1	1	1	1	8	7

When this investigation began, it was hoped that this transform would have broad application in binary image processing. First a fast algorithm was needed. This is the subject of Section 1. This search soon revealed that the transform has hidden complexity: its "neat" behavior for $N=4$ becomes "messy" with $N=8$, and with $N=16$, reversibility is no longer complete. Section 2 gives the results of work on an exhaustive enumeration of the $N=16$ case (32,768 integers). It was discovered that 896 special numbers, here called seed integers, act as singular points of the logical Walsh transform. Section 3 analyzes these integers and gives the reasons for their number and distribution. Thus instead of a nice easy-to-use transform, we have some very interesting and totally unexpected complexities.

Section 1 A Fast Algorithm?

The conventional fast Walsh transform algorithm operates only on lengths that are powers of 2, while calculation of the Walsh transform for other lengths requires multiplication by the appropriate Hadamard matrix (this "slow" way uses N^2 scalar multiplications, versus $N \log_2 N$ the "fast" way). Either way increases the storage requirement to N bits per input data point. What is desired is an algorithm that only requires 1 bit per input data point (i.e., a "Boolean" algorithm), with only moderate intermediate storage. Moreover, practical use of the logical transform, such as for binary images, requires such an algorithm to be fast (i.e., required operations per data point proportional to $\log_2 N$ rather than N).

It was originally hoped that some simple patterns in the behavior of the logical transform would provide hints for the search for a fast algorithm. For example, the Boolean function $x \otimes X$ (\otimes is addition modulo-2) is shown

in Table 1. If this simplicity occurs in the 8-bit case, it should be looked at more closely. Table 2 gives the complete 8-bit case, and patterns in the function $x \otimes X$ are more complex than in the 4-bit case.

Table 2

Hexadecimal Logical Walsh Transform For 8-Bit Case

x	X	x⊗X	x	X	x⊗X	x	X	x⊗X
80	FF	7F	90	99	9	A0	CC	6C
81	96	17	91	9F	E	A1	DE	7F
82	C3	41	92	DB	4A	A2	CF	6D
83	D7	54	93	93	0	A3	C6	6D
84	A5	21	94	BD	29	A4	ED	59
85	B7	32	95	95	0	A5	81	29
86	E7	61	96	81	17	A6	C5	6A
87	87	0	97	97	0	A7	C7	60
88	F0	78	98	F9	61	A8	FC	54
89	F6	7F	99	90	9	A9	D4	7D
8A	F3	7D	9A	D1	4B	AA	C0	6A
8B	D2	59	9B	D3	48	AB	D6	7D
8C	F5	79	9C	E1	2D	AC	E4	46
8D	B4	39	9D	B5	28	AD	F4	59
8E	E1	6F	9E	F1	6F	AE	E5	4B
8F	F7	78	9F	91	E	AF	C4	6B
B0	DD	6D	C0	AA	6A	D0	BB	6B
B1	9C	2D	C1	BE	7F	D1	9A	4B
B2	C9	7B	C2	EB	29	D2	8B	59
B3	DF	6C	C3	82	4D	D3	9B	4B
B4	8D	39	C4	AF	6B	D4	A9	7D
B5	9D	28	C5	A6	6A	D5	BF	6A
B6	CD	7B	C6	A3	6D	D6	AB	7D
B7	85	32	C7	A7	60	D7	83	54
B8	D8	70	C8	FA	32	D8	B8	70
B9	DC	65	C9	B2	7B	D9	EA	63
BA	D9	63	CA	E2	28	DA	FB	26
BB	D0	6B	CB	F2	39	DB	92	4A
BC	FD	41	CC	A0	6C	DC	B9	65
BD	94	29	CD	B6	7B	DD	30	6D
BE	C1	7F	CE	E3	2D	DE	A1	7F
BF	D5	6A	CF	A2	6D	DF	B3	6C

x	X	x⊗X	x	X	x⊗X
E0	EE	E	F0	88	78
E1	8E	6F	F1	9E	6F
E2	CA	28	F2	CB	39
E3	EA	2D	F3	8A	79
E4		48	F4	AD	59
E5	AE	4B	F5	8C	79
E6	EF	9	F6	89	7F
E7	86	61	F7	8F	78
E8	E8	Q	E8	F8	Q
E9	FE	17	F9	9F	61
EA	EA	Q	FA	C8	32
EB	C2	L9	FB	DA	26
EC	EC	Q	FC	AD	54
ED	44	49	FD	BC	41
EE		E	FE	E9	17
EF	26	9	FF	80	7F

Another pattern does stay simple in the 8-bit case: the distribution of integers that are their own transforms. There are 2 in the 4-bit case, 8 in the 8-bit case (boxed in Table 2), but 84 in the 16-bit case. These last are given in Table 3, arranged to show symmetries. Again there is increasing complexity when transform length is doubled.

Table 3

16-Bit Integers For Which $LWAL(x)=x$
Paired By Partial Complementation

807F	FF80	9009	EFF6
870F	F8F0	911E	EEEL
8241	FDBE	9249	EDB6
8421	FBDE	9429	EBD6
8356	FCA9	9333	ECCC
8536	FAC9	935F	ECA0
8661	F99E	953F	EACO
8777	F888	9555	EAAA
		9781	E87E
		9871	E78E
		9796	E869
		9976	E689
		9990	E66F
		9BD0	E42F
		9DB0	E24F
		9BD1	E42E
		9DB1	E24E
		9FF1	E00E

A44D	DBB2	B58C	CA73
A55E	DAA1	B85C	C7A3
A584	DA7B	B58D	CA72
A854	D7AB	B85D	C7A2
A7C4	D83B	B7CD	C632
AC74	D38B	BC7D	C382
AC75	D83A	BCCC	C333
AC75	D38A	BDD4	C22B

A more direct search for a Boolean function utilizes the Quine algorithm (Harrison, 1965). The 2^N possible N-bit integers map onto the vertices of the unit hypercube in the vector space of N dimensions, which is where the Walsh transform operates. Each of the N bits x_k of the logical transform is created as a separate Boolean function $x_k(x)$ and its value, 0 or 1, is put on all the 2^N vertices. All those having the value 1 are connected, and if the resultant figure is appropriately simple, we can expect to find a Boolean algorithm. The 4-bit case is shown in Figure 1, which gives the functions $x_2(x)$, $x_3(x)$, and $x_4(x)$. Here the resultant figures consist of 3 mutually perpendicular planes intersecting in a line.

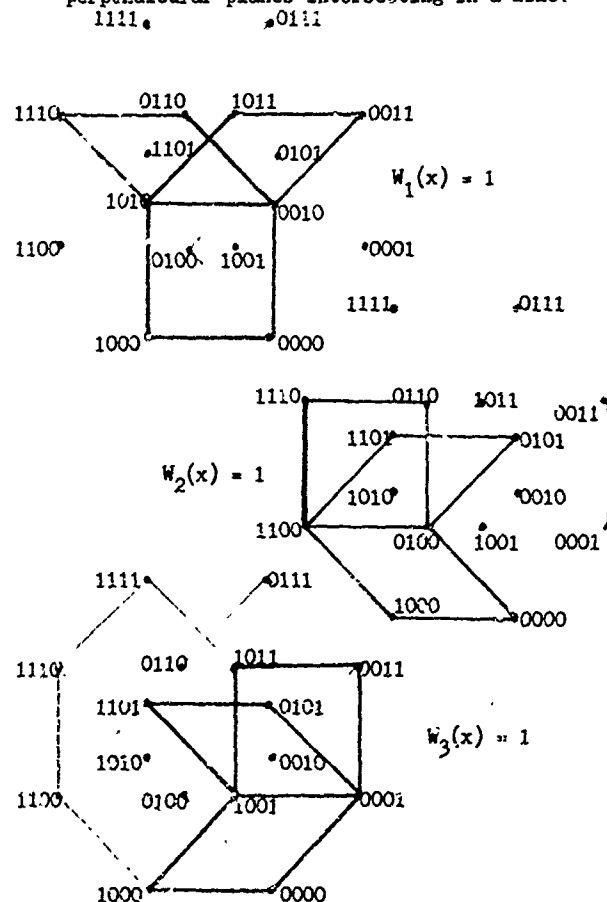


Figure 1 Quine Algorithm For N=4
Logical Walsh Transform

But in the 8-bit case, we do not find such simple patterns as, for example, 7 six-dimensional hyperplanes intersecting in a line or even in a plane, but 15 three-dimensional and 6 four-dimensional hyperplanes, all unconnected. Again, doubling the transform length makes a simple pattern become messy, further clouding the search for a fast algorithm. When the breakdown of transform invertibility was discovered for the $N=16$ case, a fast algorithm was no longer of concern; an irreversible transform destroys information and so has little chance of practical application.

Section 2

Results From $N=16$ Enumeration

After the efforts described in the preceding section, it was decided to print out by computer the logical transform function for the entire set of 16-bit integers ($2^{16} = 32,768$; $x \leq 2^{16}-1 = 65,535$). This did not require an excessive amount of computer printout, unlike the $N=32$ case ($2^{32} = 2,147,483,648$ integers). Direct visual inspection immediately proved fruitful: (1) the logical transform is no longer reversible; (2) the irreversibility takes the form of many-to-one mappings, whereby a set of 16 integers all have the same logical transform; (3) there are 896 such sets of 16 integers, for only one of which is the logical transform reversible; (4) there are 896 such special integers, paired by the logical transform function; (5) the other 15 members of each set are derived from the special integers by altering any one of its bits except the first. Because of this property, these 896 numbers are called "seed integers". Note that there are not 2048 seed integers, which would encompass the entire $N=16$ case, or even 1024, but $396 = 16008$. The reason for this is set out in detail in Section 3, and its essence lies in the Walsh transform thresholding procedure (1). The 15 derivative integers that are generated from each seed integer all have the same logical transform x . Since they differ from the seed integer by only one bit, the Walsh transform coefficients W_k each differ from the corresponding coefficients of the seed integer by ± 1 , while the corresponding logical transform bits X_k are the same. Thus it must be that the Walsh transform coefficients of the seed integer are such that a change of ± 1 or -1 will not make it cross the threshold - i.e., go from 0 to 1 or 1 to 0. Obviously this is possible only if

$$W_k \neq 0, 1 \quad k = 2, \dots, 15 \quad (2)$$

This set of 15 restrictions entirely determines the number and values of the seed integers, as we will show in Section 3.

At the beginning, however, the 896 seed integers were merely called out by computer and printed out in 112 rows of 8, whence it was quickly determined that they readily fell into 14 sets of 8 rows each. When these 8 rows of 8 seed integers each are put in hexadecimal notation, the first two bytes of each row are the same, while the last two bytes of

each "column" are the same. Thus to list all the seed integers, we need only list the 8 byte-pairs of each row and the 8 byte-pairs of each column. The 64 possible combinations are the seed integers. These 14 sets of byte-pairs are given in Table 4, where a rich structure of reciprocities and correspondences is readily apparent.

Table 4

The Seed Integers

																Second Two Bytes							
First Two Bytes																1C	2B	4D	71	8E	B2	D4	E8
81	98	A4	BB	C2	DB	E7	FE	1B	27	4E	72	8F	B1	D8	E4								
82	94	A8	BE	C1	D7	EB	FD	56	59	6C	6A	55	9A	A6	A9								
83	8C	B0	BF	C0	CF	F3	FC	1D	2E	47	74	8B	B8	D1	E2								
84	92	A1	B7	C8	DE	ED	FB	36	39	63	6C	93	9C	C6	C9								
85	84	A0	AC	D0	DF	F5	FA	35	3A	53	5C	A3	AC	C5	CA								
86	89	90	9F	E0	EF	F6	FD	11	22	44	77	88	BB	DD	EE								
87	9E	AD	B4	CB	D2	E1	F8	1E	2D	4B	78	87	B4	D2	E1								
88	91	A2	BB	C4	DD	EE	F7	12	21	48	7B	84	B7	DE	ED								
8B	9D	AE	B8	C7	D1	E2	F4	14	28	41	7D	82	BE	D7	EB								
8D	9B	A7	BF	CE	D8	E4	F2	18	24	42	7E	81	BD	DB	E7								
8E	97	AB	B2	CD	D4	E8	F1	05	0A	50	5F	A0	AF	F5	FA								
93	9C	B6	B9	C6	C9	E3	EC	03	0C	30	3F	CO	CF	F3	FC								
95	9A	A6	A9	D6	D9	E5	EA	0S	07	60	6F	90	9F	FC	F9								
A3	AC	B5	BA	C5	CA	D3	DC																

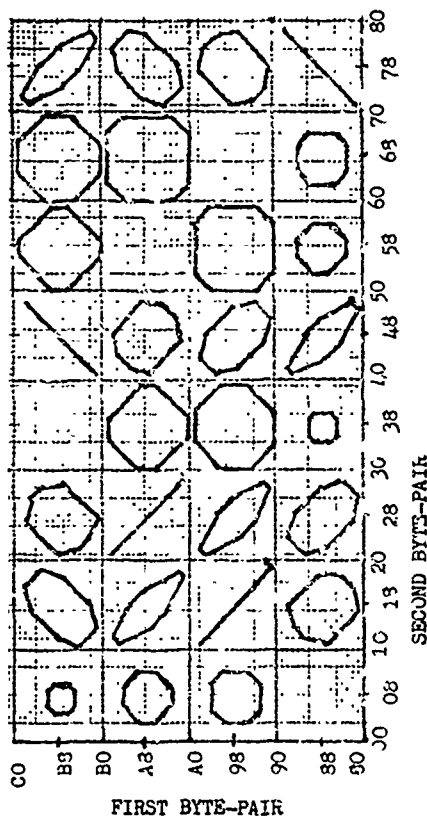


FIGURE 2
THE SEED INTEGERS

One more pattern of classifying seed integers will be presented to give a transition to Section 3. The weight of an integer is defined as the number of 1-bits in its binary representation. Seed integers fall into four weight classes: 168 with weight 6, 280 with weight 7, 280 with weight 10, 168 with weight 11. The weight -7 and weight -10 classes are closed with respect to the logical transform. Each weight class can be further subdivided into "permutation sets", which are sets of seed integers having the same numerals, in different orderings, for the second, third, and fourth bytes of their hexadecimal representation. Each group may have 1, 3, or 6 members, depending on what the numerals are. The logical Walsh transform maps an entire group onto another. Table 5 gives the permutation groups of the weight -6 seed integers, paired with the corresponding weight -11 group by the logical transform. Within a group, exchanging the second and third bytes causes the logical transform to exchange its third and fourth bytes. For example, 812B is the logical transform of D63F (and vice versa). Then 821B is the transform of D6F3, and 81B2 is the transform of D36F.

Table 6 gives the groups for weight -7, and Table 7 the groups for weight -10 seed integers. Within these groups, exchanging the second and third bytes of a seed integer does the same to the logical transform, while exchanging the second and fourth causes the third and fourth bytes of the logical transform to be exchanged.

Table 5
Permutation Groups of Weight-6 and Weight-11
Seed Integers Paired by the Logical Transform

<u>One-Member Groups</u>	
B444 : ADL7	814D : B65F
D222 : CB43	818E : F71E
E111 : 9EEE	824F : E35F
<u>Three-Member Groups</u>	
8711 : 97E7	828D : F74B
8722 : C7B7	848B : F72D
8744 : A7D7	9035 : 9D7B
87C8 : F7B7	903A : D93F
E411 : 5DED	905C : B95F
B422 : CD3D	90AC : FD1B
B488 : FD8D	A036 : CD7E
D211 : 9BEB	A039 : DC6F
D244 : AFDB	A06C : EC5F
D288 : FBDB	A09C : FD4E
E122 : 2EBF	C056 : AB7E
E144 : AEDE	C059 : BA6F
E188 : FE8E	COA6 : E3AF
<u>Six-Member Groups</u>	
812B : D63F	COA9 : F2BE
	B128 : DC9F
	D148 : BA9F
	E248 : EACF

Table 6
Permutation Groups of Weight-7 Seed Integers
Paired by the Logical Transform

<u>One-Member Group</u>	
F888 : F888	8C56 : A174
<u>Three-Member Groups</u>	
911E : 911E	8C59 : B065
922E : C11E	8C6A : E035
944E : A11D	8C9A : F124
988E : F118	9781 : 9781
A22D : C44B	912D : 941B
A44D : A44D	928B : D309
A88E : F448	948D : B509
C22B : C22B	B06A : C81D
C88B : F228	B09A : D90C
<u>Six-Member Groups</u>	
8356 : 8356	C28E : E30A
8359 : 9247	ECA0 : ECA0
836A : C217	A12E : C41E
839A : D306	A18B : D60C
	A24B : C24D
	A278 : C847
	A48E : E50C

Table 7
Permutation Groups of the Weight-10 Seed
Integers Paired by the Logical Transform

<u>One-Member Group</u>	B78B : D78D
8777 : 3777	B1BE : DD1E
<u>Three-Member Groups</u>	B2BD : DB4B
8BB7 : D772	E2DE : EB4E
SDD7 : D772	935F : 935F
8EE7 : E771	9C5F : E17D
BBB4 : DDD2	C95F : B27E
BDD4 : BDD4	93AF : D71B
BEE4 : EDD1	A39F : D74E
DBB2 : DBB2	AC9F : F56C
DDE2 : EBB1	9CAF : F539
EEE1 : EEE1	C9AF : F63A
<u>Six-Member Groups</u>	A36F : C65F
8BDE : F356	AC6F : E47D
B8DE : F95C	CA6F : E27B
E88D : FA6C	E78E : E78E
D8BE : F93A	FCA9 : FCA9

Section 3

Analysis of the Seed Integers

In this section we will show how the defining property of a seed integer, given by (2), generates the configurations of Tables 5, 6, and 7.

We will give an analytic enumeration of all the possible bit-configurations. But first there are two equivalences which reduce the amount of enumeration. These are partial and full complementation. Complementation changes 1-bits to 0 and 0-bits to 1. It is partial if the first 1-bit is unchanged. We will prove that the partial and full complements of a seed integer also fulfill (2).

For partial complementation, take the Walsh transform vectors of an integer x and of its partial complement \bar{x} . Their vector sum has the form

$$W(x) + W(\bar{x}) = 17, 1, 1, \dots, 1$$

Thus if $W(x)$ does not equal 0 or 1, neither does $W(\bar{x})$. Thus it is only necessary to enumerate the integers with an even number of 1-bits, and their partial complements will cover the entirety of the integers with an odd number of 1-bits. When a seed integer has an even number of 1-bits, its full complement fulfills (2), as we see from forming the vector sum

$$W(x) + W(\bar{x}) = 16, 0, 0, \dots, 0$$

If $W(x)$ consists entirely of even nor zero integers, so does $W(\bar{x})$. Thus we need only check integers with 2, 4, 6, or 8 1-bits, with the previous restriction of $x_1 = 1$ removed, and all the other bit configurations are generated by partial or full complementation.

Weight 2 If both 1-bits are in the first half of the binary number, or both in the second half, there will be 4 Walsh coefficients that zero out because the two 1-bits are weighted by +1 and -1. If there is one in each half, then $W_8(x) = 0$. We will call these reasons Rule A and Rule B, respectively.

Weight 4 If all 4 1-bits are in either half of the binary number, Rule A applies. If there are 2 1-bits in either half, Rule B applies. If we divide the bit-configuration into four quarters and designate a configuration by their weights, the remaining weight-4 cases are (2,1,1,0), (2,1,0,1), (1,0,2,1), (1,0,1,2), (0,1,2,1) and (0,1,1,2) (collectively designated by [2,1,1,0]) and (3,0,1,0), (0,3,1,0), (3,0,0,1), (0,3,1,0), (1,0,3,0), (0,1,3,0), (1,0,0,3), and (0,1,0,3) (collectively designated by [3,1,0,0]). The cases [2,1,1,0] are eliminated by at least one Walsh coefficient for which the weight-2 quarter contributes zero and the other two weight-1 quarters contribute +1 and -1, for a zero total. This is called Rule C. The cases [3,1,0,0] are eliminated by at least one Walsh coefficient for which the weight-3 quarter contributes ± 1 , and the weight-1 quarter contributes ± 1 .

Weight 8 Rule A eliminates all integers with the 8 1-bits in either half of the binary number. Rule B eliminates all those with 4 1-bits in both halves. Rule C eliminates the cases [4,3,1,0], [4,2,2,0] and [3,2,2,1]. That is, there is at least one Walsh coefficient that zeros out the even weight quarters, and cancels the remaining two odd-weight quarters. For the cases [4,2,1,1] and [3,3,2,0] at least two of the even Walsh coefficients are zero. In [4,2,1,1], the weight-4 and weight-2 quarters will zero out and the two weight-1 quarters will make opposite contributions. Similarly, in [3,3,2,0] the two weight-3 quarters will make opposite contributions and the weight-2 quarter will zero out. This is Rule D.

Weight 6 Rule A eliminates all integers with the 6 1-bits in either half. Rule B eliminates all those with 3 1-bits in both halves. Rule C eliminates the cases [3,2,1,0] and [4,1,1,0]. This leaves the cases [2,2,2,0] and [3,1,1,1] some of which can be seed integers. From these all the seed integers can be derived.

First the simpler case, [2,2,2,0]. Each quarter L_i has a 4-component Walsh transform vector: $W(L_i) = (w_1, w_2, w_3, w_4)$, where each component weights the 4 bits of L_i accordingly: $w_1, +++$; $w_2, +-+$; $w_3, ++-$; $w_4, +--$. Table 8 gives the weight-2 hexadecimal numerals, paired by complementation, their bit-configurations, and their length-4 Walsh transform coefficients. Now the length-16 Walsh transform is formed by the concatenation four 4-component vectors, each of which is a combination of the length-4 Walsh transforms of the four quarters of x :

$$\begin{aligned}
 W(x) &= (W(L_1) + W(L_2) + W(L_3) + W(L_4); \\
 &\quad W(L_1) - W(L_2) + W(L_3) - W(L_4); \\
 &\quad W(L_1) + W(L_2) - W(L_3) - W(L_4); \\
 &\quad W(L_1) - W(L_2) - W(L_3) + W(L_4)).
 \end{aligned} \quad (3)$$

Table 8

Weight-2 Hexadecimal Numerals								
x	x ₁	x ₂	x ₃	x ₄	w ₁	w ₂	w ₃	w ₄
3	0	0	1	1	2	0	-2	0
C	1	1	0	0	2	0	2	0
5	0	1	0	1	2	-2	0	0
A	1	0	1	0	2	2	0	0
6	0	1	1	0	2	0	0	-2
9	1	0	0	1	2	0	0	2

Applying this to Table 8, we see that for the coefficients $W_1(x)$, $W_6(x)$, $W_{10}(x)$, $W_{14}(x)$ to be non-zero, we must have either 5 or A and not both, in one of the quarters. For the coefficients $W_3(x)$, $W_7(x)$, $W_{11}(x)$, $W_{13}(x)$ to be non-zero, we must have either 3 or C, and not both, in another of the quarters. Finally, for the coefficients $W_4(x)$, $W_8(x)$, $W_{12}(x)$, $W_{13}(x)$, either 6 or 9, not both, in the other weight-2 quarter. The total number of integers satisfying these conditions is calculated as follows: the zero may go in any of the 4 quarters, any of the 6 weight-2 numerals may go in another quarter, any of the 4 remaining in the next quarter, and either of the last 2 in the final quarter; $4 \cdot 6 \cdot 4 \cdot 2 = 192$. Under the constraint $x_1 = 1$, the total is $3 \cdot 3 \cdot 4 \cdot 2 = 72$. These fall into the 168 weight-6 seed integers of Table 5. Their partial complements fall into the 168 weight-11 seed integers of Table 5. The remaining $192 - 72 = 120$ integers with $x_1 = 0$ are fully complemented, and fall into the 280 weight-10 seed integers of Table 7. These 120 weight-7 seed integers are in turn partially complemented and placed with the 280 weight-7 seed integers of Table 6.

Next the more complex [3,1,1,1] case. Table 9 gives the weight-1 and weight-3 hexadecimal numerals, paired by complementation, their bit configuration, and their length-4 Walsh transforms. The members of each pair are designated by type t_1 according to the bit-position where the 1 or 0 is placed. The subscript 1 indicates which of the four quarters of x the particular numeral is placed. There are two different possibilities: two quarters have the same type, or have different types. For the former, assume one of these quarters is the weight-2 quarter. Then in (3), the contribution of this pair to $W(x)$ is (4,0,0,0) when their length-4 Walsh coefficients $W(L_1)$ and $w(L_1)$ are added, and it is (± 2 , ± 2 , ± 2 , ± 2) when they are subtracted. Now look at the remaining two quarters, L_k and L_l , both of weight 1. When $t_k = t_l$, then contributions to $W(x)$ are (± 2 , ± 2 , ± 2 , ± 2) when added and (0,0,0,0) when subtracted. Thus the 4-component vectors concatenated into $W(x)$ all have the form (6 or 4, ± 2 , ± 2 , ± 2) or (± 2 , ± 2 , ± 2 , ± 2), so that $W_k(x) \neq 0$ for all k . However, when $t_k \neq t_l$, their contributions to $W(x)$ contain at least one 0, and when combined with

(4,0,0,0) gives $W_k(x) = 0$ for some k . Thus the bit configuration with $t_1 = t_j$, $t_k \neq t_l$ generates seed integers and $t_1 = t_j$, $t_k \neq t_l$ cannot. These seed integers enumerate as follows: the weight-3 quarter may be in any of 4 places, and may be any of 4 numerals; this numeral determines the numeral of the same type, which may be in any of 3 places; the remaining two quarters are the same, and may be occupied by any of the 3 remaining weight-1 numerals. When all three weight-1 quarters are the same, there are 16 more combinations. The total is $4 \cdot 4 \cdot 3 \cdot 3 + 4 \cdot 4 = 160$. Under the constraint $x_1 = 1$, the total is $3 \cdot 10 + 3 \cdot 4 + 3 \cdot 6 = 60$. These fall into the 168 weight-6 seed integers of Table 5, while their partial complements fall into the 168 weight-11 seed integers of Table 5. The remaining $160 - 60 = 100$ with $x_1 = 0$ are fully complemented and included in the 280 weight-10 seed integers of Table 7. These 100 weight-10 seed integers are in turn partially complemented and included in the 280 weight-7 seed integers of Table 7.

Table 9

Weight-1 and Weight-3 Hexadecimal Numerals

x	x ₁	x ₂	x ₃	x ₄	w ₁	w ₂	w ₃	w ₄
1	0	0	0	1	1	-1	-1	1
E	1	1	1	0	3	1	1	-1
2	0	0	1	0	1	1	-1	-1
D	1	1	0	1	3	-1	1	1
4	0	1	0	0	1	-1	1	-1
F	1	0	1	1	3	1	-1	1
8	1	0	0	0	1	1	1	1
7	0	1	1	1	3	-1	-1	-1

Finally, we examine the case [3,1,1,1] bit configurations with the initial constraint of $t_1 \neq t_j$. We already know that $t_k \neq t_l$, since the $t_k = t_l$ case is the mirror of $t_1 = t_j$, $t_k \neq t_l$ which as we have shown above fails to produce seed integers. Thus all four quarters are of different type. Consider the Walsh coefficients $W_5(x)$, $W_9(x)$, $W_{13}(x)$. Applying (3) to Table 9, we see that these coefficients all have the values $\pm((3+1) - (1+1)) = \pm 2$. All the other coefficients (except of course for $W_1(x)$) have the form $\pm(1+1+1-1) = \pm 2$. Thus $W(x) \neq 0$ for all k . These enumerate as follows: the weight-3 quarter may be any one of 4 numerals, placed in any one of 4 places; the remaining weight-1 quarters are selected in $3 \cdot 2 \cdot 1$ possible combinations. The total is $4 \cdot 4 \cdot 3 \cdot 2 = 96$. Under the constraint $x_1 = 1$, the total is $3 \cdot 3 \cdot 2 + 3 \cdot 3 \cdot 2 = 36$. These and their partial complements fall into Table 5, for a grand total of $72 + 60 + 36 = 168$. The remaining $96 - 36 = 60$ with $x_1 = 0$ are fully complemented, then partially complemented, to complete the enumeration of the weight-10 and weight-7 seed integers, respectively. Their grand total is $120 + 100 + 60 = 280$, which agrees with the computerized enumeration.

Conclusion

The last section showed how the 4-bit Walsh transform comes into the determination of the seed integers. Thus the seeming simplicity of the $N=4$ case actually contributes to the complexity of the $N=16$ case. That is, the properties of the 4-bit case are the reasons there are only 896 seed integers, instead of a much "neater" 1024 or 2048. Thus as transform reversibility breaks down, it does so in a coherent manner, but with gaps, so to speak, in the coherence: the blank squares in Figure 2 attest to this. This fits right in with the results of Section 1, which showed that doubling the transform length greatly increases the "messiness" of the transform. A random sampling of the $N=32$ case shows a continuation of this trend: a high percentage of the integers are irreversible, and the transform can be done many times without it "settling down" to a reversible integer. Furthermore, the analysis derivation of seed integers is beset with difficulties: (1) there are 31 simultaneous inequalities, (2) sometimes more than one bit can be varied without changing the logical transform.

Besides the $N=32$ case, research can be continued in other directions: (1) defining a logical transform for Hadamard matrices other than 2^n ; (2) defining other transforms by altering the geometry of Figure 2 (remember that the analysis of Section 3 gives no reason for these octagons); (3) using a different procedure to derive X from W , in an attempt to restore reversibility. While only the last proposal can have any practical application in binary data processing, the other two are of general interest in Walsh-function research and Boolean algebra. The increasing "messiness" of the logical transform may have interest in the philosophy of mathematics, which is usually occupied with "well-behaved" phenomena.

References

- Searle, N. H., "A Logical Walsh Transform", Proceedings of the 1970 Symposium on Walsh Functions.
- Parkyn, W. A., "Digital Image Processing Aspects of the Walsh Transform" *ibid.*
- Harrison, M. A., Introduction to Switching and Automata Theory, McGraw Hill 1965

APPROXIMATION ERRORS OF A WALSH SERIES

C.K. Yuen, Ph.D. student
Basser Department of Computer Science
University of Sydney, Sydney, Australia

Function representation by Walsh series is a profitable exercise only when it is both accurate and economical. By accuracy we mean that the series must satisfy some error criterion. If the function being represented is a random signal the accuracy is usually measured by the mean square deviation while numerical applications would require that the maximum absolute error be below some upper limit. Economy demands that the Walsh series contains less terms than the amount of data necessary to specify the original function. Thus the amount of information contained in the original function is compressed into a smaller set of numbers.

The common technique for choosing the terms to be retained in Walsh serieses is threshold sampling: any Walsh function whose coefficient is less than some threshold is dropped from the series, or equivalently, assigned coefficient zero. This results in the retention of terms with largest coefficients and gives an approximation that is optimal in the least square sense.

There are, however, two disadvantages. One is that it is impossible to know beforehand which terms will be retained. This is an extremely serious problem for information transmission problems. For example, to use Walsh functions in picture transmission it is necessary to compute all the Walsh transforms, select those with largest magnitudes and send these values over the channel with their indices since the receiving end has no knowledge which Walsh transforms would be selected. This results in an increase in required bandwidth partially defeating the purpose of using Walsh transforms. The second is the difficulty of error analysis, since threshold sampling is similar to rounding by setting to zero any number that is smaller than some limit. When a large number of terms are dropped, the theoretical upper bound on the total error is simply the number of terms times the threshold. This gives an enormous upper bound which is totally unrealistic.

The lack of theoretical analysis of errors is no headache for information transmission applications since trial and error experimentation is considered quite respectable and has indeed produced many interesting results. Nevertheless it is interesting to know that it is in fact quite easy to derive fairly realistic upper bounds on the approximation errors of Walsh serieses if we group Walsh functions by their degrees, to be defined below. When we neglect a group of terms with the same degree, the errors are related and do not accumulate like rounding errors.

In selecting Walsh functions by their degrees, a term is retained in the series if it has the appropriate degrees, regardless of its magnitude. Thus, a data compression scheme based on degrees does not always choose terms of largest magnitudes. However, though our selection scheme is different from threshold sampling, the performances of the two tend to be very similar, for numerical applications at least. Thus, prediction of the performance of a Walsh series whose terms were selected by their degrees is also a good prediction for the accuracy of a series whose terms were selected by threshold sampling.

To derive upper bounds on approximation errors we could use an "integration by parts" technique which have been employed in a previous work[1] studying upper bounds on single Walsh transforms. In some cases upper bounds can be derived by direct manipulation of certain integrals. To avoid copying tedious mathematical arguments only the main results will be shown. Detailed proofs may be found in [1] and in a subsequent paper [2]. We shall also give several numerical examples. Finally, extending the error analysis technique to information transmission problems will be discussed.

Mathematical Introduction

Let us have

$$t = (0.t_1t_2 \dots t_k \dots)_2 \in [0,1)$$

$$i = (\dots i_k \dots i_2i_1)_2$$

The k th Rademacher function is

$$R_k(t) = (-1)^{t_k}$$

and the i th Walsh-Paley function

$$\text{pal}(i,t) = \prod_{k=1}^{\infty} [R_k(t)]^{i_k}$$

The degree of $\text{pal}(i,t)$ is defined to be the position of the first 1 in i :

$$d(i) = \min \{k: 2^k > i\},$$

and its subdegree position of the second 1:

$$d'(i) = \min \{k: 2^k > i - 2^{d-1}\}.$$

The total number of 1's in i is called the rank of $\text{pal}(i,t)$:

$$r(i) = \sum_{k=1}^{\infty} i_k$$

and its negligibility is defined as

$$p(i) = \sum_{k=1}^{\infty} i_k k + r(i).$$

Given a function $x(t)$ we define its Walsh transform as

$$F_i = \int_0^1 f(t) \text{pal}(i, t) dt.$$

We proved elsewhere that for all $k \leq r(i)$

$$F_i = (-1)^k \int_0^1 f^{(k)}(t) \int_0^1 \dots \int_0^{t_{k-1}} \text{pal}(i, t_k) dt_k \dots dt_1 dt. \quad (1)$$

That is, in computing F_i we can differentiate $f(t)$ up to $r(i)$ times and simultaneously integrate $\text{pal}(i, t)$ the same number of times, as in integration by parts. This gives the following upper bound on magnitude of F_i [1]:

$$|F_i| \leq \max |f^{(r)}(t)| 2^{-p(i)}. \quad (2)$$

Thus if $p(i)$ is large then F_i has small magnitude so can be neglected. This is the reason for the name "negligibility".

Truncation errors

Polyak and Schreider proved that [3]:

$$\sum_{i=0}^{2^n-1} \text{pal}(i, t) \text{pal}(i, T) = B^n(t, T)$$

where $B^n(t, T) = 0$ except when $[2^n T] = [2^n t]$, in which case

$$B^n(t, T) = 2^n.$$

We shall consider t to be fixed. Then $B^n(t, T)$ is zero if T is less than $2^{-n} [2^n t]$ or greater than $2^{-n} [2^n t + 1]$ and is constant in between. It is thus a block pulse of height 2^n and width 2^{-n} . For convenience we shall denote $2^{-n} [2^n t]$ by t^n , $2^{-n} [2^n t + 1]$ by t^{n+} and $2^{-n} [2^n t + 1]$ by t^{n++} .

We can now derive an upper bound on the error caused by dropping terms

$$F_{2^N} \text{ to } F_{2^{N+1}-1}$$

from the Walsh series. We have

$$f(t) = \sum_{i=0}^{\infty} F_i \text{pal}(i, t).$$

Neglecting 2^N terms causes error

$$e(N) = \sum_{i=2^N}^{\infty} \text{pal}(i, t) \int_0^1 \text{pal}(i, T) f(T) dT$$

For each i in the above expression $i_{N+1} = 1$ so we have

$$\text{pal}(i, t) = R_{N+1}(t) \text{pal}(j, t)$$

with $0 \leq j < 2^N$. Thus

$$e(N) = R_{N+1}(t) \int_0^1 R_{N+1}(T) L^N(t, T) f(T) dT.$$

As the integrand is non-zero only between t^N and t^{N++} and since R_{N+1} is 1 for first half of the interval and -1 in second half, we get

$$\begin{aligned} |e(N)| &= |2^N \int_{t^N}^{t^{N++}} f(T) dT - 2^N \int_{t^{N+}}^{t^{N++}} f(T) dT| \\ &= |2^N \int_{t^N}^{t^{N+}} \int_{T+2^{-N-1}}^T f'(T_1) dT_1 dT| \\ &\leq \max |f'(t)| 2^{-N-2}. \end{aligned}$$

We first observe that each of the 2^N terms dropped has degree $N+1$. We then note that the first term among them, F_{2^N} , has upper bound

$$|F_{2^N}| \leq \max |f'(t)| 2^{-N-2},$$

according to (2). Thus the upper bound on error caused by dropping 2^N terms is the same as the bound on one term. We see that errors do not accumulate if we neglect a group of terms with a common degree.

Truncating the series at 2^N causes error

$$\begin{aligned} |e_t| &\leq |e(N)| + |e(N+1)| + \dots \\ &\leq \max |f'(t)| 2^{-N-1}. \end{aligned}$$

It may be observed that truncation at 2^N amounts to neglecting all F_i with degree larger than N . Note also that truncation error occurs only when we choose 2^N to be less than the number of input data. For example, if the argument t is given as an n bit number, then there are only 2^n different values it can take on, hence there are only 2^n input numbers. If we use a 2^n term Walsh series the representation is exact and there is no truncation error.

Data compression errors

The truncated Walsh series contains 2^N terms, which can be gathered into $N+1$ groups, each with a different degree. Thus the group

$$F_{2^{d-1}} \text{ to } F_{2^d-1}$$

contains all terms with degree d .

Suppose we take the group with degree $d+1$ and choose to neglect all terms in it that have subdegrees greater than d' , the

error caused is

$$e(d, d') = \sum_{i=2^d+2^{d'}}^{2^{d+1}-1} \text{pal}(i, t) \int_0^1 \text{pal}(i, T) f(T) dT.$$

This can be put into

$$R_{d+1}(t) \int_0^1 [B^d(t, T) - B^{d'}(t, T)] R_{d+1}(T) f(T) dT$$

and has upper bound

$$|e(d, d')| \leq \max f''(t) 2^{-d-d'-5} (1-2^{d'-d})$$

We may note that the first term among the group of terms neglected is $F_{2^d+2^{d'}}$ which obeys

$$|F_{2^d+2^{d'}}| \leq \max f''(t) 2^{-d-d'-4}.$$

Thus the error caused by neglecting terms $F_{2^d+2^{d'}}$ to $F_{2^{d+1}-1}$ is less than twice the upper bound on one term, showing again that errors do not accumulate.

The above argument shows that we can select terms to be retained in a Walsh series according to their degrees and sub-degrees. Any term with degree above some pre-set limit is neglected (truncation). Then each term with the correct degree but a subdegree exceeding some number, which may be dependent on the degree, is dropped to achieve data compression. That is, for each of the $N+1$ groups we specify a maximum subdegree. The error caused by neglecting the set of terms in each group can be estimated by examining its upper bound, and the total approximation error estimated by summing all the upper bounds. The total number of terms kept in the series can also be calculated immediately. We can thus decide whether the Walsh series is sufficiently accurate and economical without having to actually compute it.

Two questions must be answered. First, do the upper bounds give accurate estimates of actual errors? Second, is the performance of this scheme for selecting terms as good as threshold sampling? We note that, for two series with the same number of terms, the one chosen by threshold sampling gives less mean square error, while its absolute error may be unevenly spread over the function domain so that the maximum error may not be less, though for practical cases it is not far from the maximum of the best series. We shall see later that the error upper bound we derived is a good though far from perfect indicator of the real error. We shall also see that, if the function being approximated has sizable high order derivatives then a series selected by threshold sampling contains many high rank terms that would not normally be selected if selection by degrees is used. If size of derivatives does not increase rapidly with each differentiation then threshold sampling and selection by degrees tend to choose the same terms. It is

not hard to see why. Equation (2) indicates that, unless $f(t)$ has large high order derivatives F_i is small if its rank is large, so it is unlikely to be chosen in threshold sampling. On the other hand selection by degrees tends to choose terms of small second degrees, which generally means that i contains few 1's so its rank is usually small. So both schemes tend to select low rank terms, and so have similar performance.

Examples

1. We first give an example to illustrate the misleading results one gets by applying straightforward error analysis to threshold sampling.

To be useful a Walsh series must contain only a small number of terms. To obtain such a series we can either choose a small N , compute a truncated series of 2^N terms and drop a small part of them; or we choose a large N and neglect a major proportion of the 2^N terms. According to straightforward analysis the error caused by dropping terms is proportional to the number of terms neglected, so the second alternative would give enormous data compression error. This was the conclusion reached by Polyak and Schreider [5] in their study on function approximation using Walsh series, hence they truncated their examples at $2^N=64$ and used data compression ratio of around 50%. As a result their Walsh series approximations are little better than tabulation. Their series for $f(t) = \sin(\frac{1}{2}\pi t)$ contains 25 terms and has error about 0.008.

We chose, instead, $N=8$ and dropped $F_{2^d+2^{d'}}$ to $F_{2^{d+1}-1}$ for the following pairs of (d, d') :

$$(d, d') = (7, 1), (6, 1), (5, 2), (4, 2), (3, 2).$$

Total number of terms kept is 24, the truncation error is bounded by .0015 and data compression error by .0040. The computed errors are a little less, while the error of 24 term series selected by threshold sampling is slightly smaller yet. If t is given to 8 bits only then truncation error is eliminated and the approximation is accurate to 2^{-8} .

2. We took $f(t) = \exp(-t)$, $N=12$, and dropped terms for pairs of (d, d') :

$$(11, 0), (10, 0), (9, 1), (8, 1), (7, 2), (6, 2), (5, 3),$$

as well as terms F_{11010} to F_{11111} , with total data compression error bounded by 0.0017. The actual error was computed to be 0.00156. The number of terms kept is 48. For the same function a series of 49 terms, selected by neglecting all terms with absolute values less than 0.00005, has maximum error of 0.00101. 43 terms are common to both series but 6 terms with magnitudes twice or three times of threshold are not included in

the first series.

We see that, in above two examples threshold sampling performs better than selection by degrees, though error bounds of the latter still provides a rough indication as to the accuracy of the former. If we had treated error caused by threshold sampling as rounding error, we would have derived an upper bound by multiplying the threshold by the number of terms dropped, which would give, for example 2,

$$|\text{error}| \leq 0.2$$

which is quite unrealistic.

3. The function $x^{\frac{1}{2}}$ has an infinite second derivative at the origin. As the error bound for selection by degrees scheme depends on the second derivative, we would expect a Walsh series approximation to perform poorly. Also, because the function has very large high order derivatives, its high rank Walsh transforms would be sizable, hence would be selected by threshold sampling. Indeed, we found that, even if we set the threshold as high as 0.0002 the number of terms is still no less than 151, while the error at $t=0$ is 0.05. (maximum error for $\frac{1}{4} \leq t \leq 1$ is 0.00132)

However, we may note that we can restrict the argument to between $\frac{1}{4}$ and 1 since any other value can be treated by shifting it right or left two bits at a time since $f(\frac{1}{4}t) = 2^2 f(t)$. So although the Walsh series is defined for $t \in [0,1)$ its error for t between 0 and $\frac{1}{4}$ is immaterial.

This fact is taken advantage of by using the following function

$$f(t) = \begin{cases} 0.5+t & 0 \leq t < 0.25 \\ t^{\frac{1}{2}} & 0.25 \leq t < 1 \end{cases}$$

For we have $f''(t) = 0$ between 0 and $\frac{1}{4}$, so that

$$\max |f''| = |f''(\frac{1}{4})| = 2,$$

and the error would fall within acceptable limits. N was chosen to be 12 and

$$(d, d') = (11, 1), (10, 1), (9, 1), (8, 1), (7, 1), (6, 3), (5, 3).$$

Total number of terms kept is 58, and maximum error is 0.00119. A second series found using threshold sampling has the same number of terms with error 0.00116. The two series contain 51 common terms.

While the error upper bound, in this instance, gave a poor estimate (0.003), the knowledge that it depends on the second derivative greatly assisted in finding a way around the difficulty we would have encountered if we had attempted a straightforward Walsh series expansion.

Discussion

The examples we presented should make clear that as technique for writing mathematical subroutines Walsh series cannot comp-

ete with polynomial approximation as the latter requires far less data storage, memory access during evaluation and total computing time. However, the hardware implementation of Walsh series appear economical since the evaluation of a Walsh series needs only fixed point additions. The operations involved, fetch and add repeatedly, are simple and can be speeded up at little cost.

We finally mention that it is also possible to estimate the signal power loss due to bandwidth compression if we assume that Walsh transforms are selected by their degrees. Given random variable $x(t)$ with autocorrelation function $V(t)$ we have the i th Walsh power spectral element

$$X_i = \iint \text{pal}(i, T_1) \text{pal}(i, T_2) V(T_1 - T_2) dT_1 dT_2$$

$$\text{And } e^2(d, d') = \sum_{i=2^d+2^{d'}}^{2^{d+1}-1} X_i =$$

$$\int R_{d+1}(T_1) \int [B^d(T_1, T_2) - B^{d'}(T_1, T_2)] R_{d+1}(T_2) V(T_1 - T_2) dT_2 dT_1$$

This finally leads to

$$|e^2(d, d')| \leq 2^{-d-2} (1 - 2^{d'-d}) \int_{-2^{-d'}}^{2^{-d}} |R''(t)| dt.$$

The reader is cautioned that the above upper bound is unlikely to be as good as those on truncation and data compression errors of a smooth function. Also, it is possible that selection by degrees, whose error is a weighted integral of the function being approximated, causes a type of distortion unacceptable for random signals. Only experiments would give final answers.

We conclude by saying that, although Walsh functions are discrete and binary, mathematical analysis of Walsh series errors are at least possible and trial-and-error experimentation, though still necessary, is not the only thing to do.

References

1. C.K. Yuen, "Upper bounds on Walsh transforms", Basser Computing Department Tech. Report No. 66, University of Sydney, Sept. 71
2. C.K. Yuen, "Approximation errors of a Walsh series", to appear.
3. B.T. Polyak and Y.A. Schreider, "The application of Walsh functions in approximate calculations", Problems in Theory of Mathematical Machines, Collection II, Fizmatgiz, Moscow, 1966, pp. 175-190.

Acknowledgement

Thanks are due to Dr. H.R. Hwa for supervising the research which led to this work.

THE HILBERT SCHMIDT DECORRELATION MEASURE FOR FOURIER AND WALSH TRANSFORMS

J. Pearl
Associate Professor of Engineering
University of California, Los Angeles
Los Angeles, California

ABSTRACT

The paper summarizes results of our investigation into the merit of Fourier and Walsh processing of random signals. It has been often conjectured that the popularity of Fourier processing stems from the fact that the Fourier coefficients of stationary signals are uncorrelated and that the Walsh coefficients are "almost" uncorrelated. Our results demonstrate that the Fourier coefficients are "almost" uncorrelated while the Walsh coefficients are "rarely" uncorrelated. By the word "almost" we mean that as the data-block size (N) increases the coefficients become uncorrelated and the degradation in performance due to residual correlation goes to zero. The word "rarely" states that the performance degradation does not vanish for very large N , except in very few pathological cases (white noise, perfect correlation, etc.).

SUMMARY OF PREVIOUS RESULTS

In earlier papers (see references 1, 2, 3) we have established the following results:

1. By the words "spectral signal processing" we invariably imply that after the spectral representation of a signal is computed its components are to be handled independently of each other in the next processing stage.
2. In Filtering and Coding applications, and under a mean-square-error fidelity criterion the optimal spectral representation is the Karhunen-Loève expansion (rendering the spectral components uncorrelated).
3. If $\{\lambda_i\}$ $i = 1, 2, \dots, N$ are the eigenvalues of the input covariance matrix, then the performance of the Karhunen-Loève processing is given in the form

$$P_{K-L} = \frac{1}{N} \sum_{i=1}^N F(\lambda_i) \quad (1)$$

where $F(\cdot)$ is a convex function.

4. When using a unitary transform U other than the Karhunen-Loève, the optimal performance that can be achieved is given by

$$P_U = \frac{1}{N} \sum_{i=1}^N F(\sigma_i^2) \quad (2)$$

where σ_i ($i = 1, 2, \dots, N$) are the variances of the spectral coefficients (or the diagonal elements of the covariance matrix in the U representation).

5. Thus, the degradation in performance due to incomplete decorrelation is given by:

$$\Delta R = \frac{1}{N} \sum_{i=1}^N [F(\lambda_i) - F(\sigma_i^2)] \quad (3)$$

SUMMARY OF CURRENT RESULTS

1. Rather than calculating equation (3) for every possible covariance matrix and for every performance function F we attempted to find a condition that would guarantee

$$\lim_{N \rightarrow \infty} \Delta R = 0 \quad (4)$$

for all weakly stationary signals. Such condition can be given in terms of the Hilbert-Schmidt norm

$$|A|^2 = \frac{1}{N} \sum_{i,j} |A_{ij}|^2 \quad (5)$$

2. If T_N is a Toeplitz covariance matrix and C_N is a circulant approximation of T_N obtained by ignoring all off-diagonal elements in its Fourier representation, then

$$\lim_{N \rightarrow \infty} |\underline{T}_N - \underline{C}_N|^2 = 0 \quad (6)$$

(See reference 4 for proof.)

3. If (6) is satisfied and \underline{T}_N is bounded then (4) is guaranteed (ref. 5). This implies that Fourier processing is asymptotically optimal for stationary signals.
4. For large N ΔR is proportional to $|\underline{T}_N - \underline{C}_N|$ in both Coding and Filtering (ref. 4).
5. For finite order Markov processes, $|\underline{T}_N - \underline{C}_N|$ approaches zero like $1/\sqrt{N}$ (ref. 4).
6. The last two statements imply that the performance achievable by Fourier Processing approaches the optimal like $1/\sqrt{N}$.
7. The Walsh transform is not asymptotically optimal for stationary signals. The performance difference between Walsh and Karhunen-Loève processing does not vanish except in a few rare cases. This can be established by observing the behavior of $|\underline{T}_N - \underline{D}_N|$ where \underline{D}_N is the dyadic approximation of \underline{T}_N (obtained by ignoring the off-diagonal elements of $\underline{W} \underline{T}_N \underline{W}^{-1}$). In most cases $|\underline{T}_N - \underline{D}_N|$ reaches a non-zero asymptotic value for large N , as exemplified by the moving-average process in figure 1.

CONCLUSIONS

In signal processing applications of stationary inputs (or asynchronous inputs) the Fourier Transform offers a definite advantage over the Walsh Transforms. The Fourier Transform is asymptotically optimal for all weakly stationary signals, while the Walsh Transform remains suboptimal even for very long blocks.

REFERENCES

- (1) Pearl, J., "Walsh Processing of Random Signals", Proceedings of the Symposium on the Applications of Walsh Functions, Washington, D.C., April 1971, pp. 137-141.
- (2) Pearl, J., "Basis-Restricted Transformations and Performance Measures for Spectral Representations", IEEE Transactions on Information Theory, Vol. IT-17, No. 6 (November 1971), pp. 751-753.

- (3) Pearl, J., Andrews, H. C., and Pratt, W.K., "Performance Measures for Transform Data Coding", to be published in IEEE Transactions in Communication Technology, June 1972.
- (4) Pearl, J., "On the Finite Fourier Transform of Stationary Time Series", to be published.
- (5) Gray, R. M., "Toeplitz and Circulant Matrices: A Review", Stanford University Report SU-SEL-71-032, June, 1971.

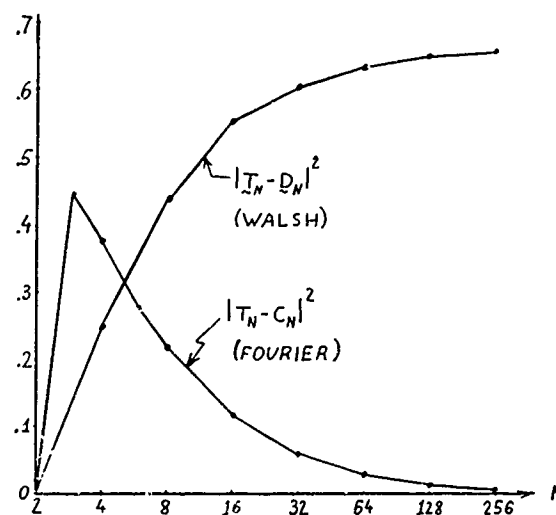


FIGURE 1. HILBERT SCHMIDT NORM FOR FOURIER (\underline{C}_N) AND WALSH (\underline{D}_N) TRANSFORMS OF A MOVING-AVERAGE PROCESS.

This work was supported by the Office of Naval Research, Mathematical and Information Science Division, Contract No. N00014-71-0200-4038.

WALSH SERIES TO FOURIER SERIES CONVERSION
K.H. Siemens and R. Kitai
Department of Electrical Engineering
McMaster University
Hamilton, Ontario, Canada

Abstract

The coefficients of the Walsh series of a function can be used to derive the corresponding Fourier series. The conversion equation for each Fourier coefficient is in the form of an infinite summation of products of constants and the Walsh coefficients. Truncation of the conversion equation generally gives rise to errors. It is shown that precise evaluation of the Fourier coefficients in terms of Walsh coefficients is possible if the signal is either sequence-limited or frequency-limited. A dual relationship holds for Fourier series to Walsh series conversion.

Introduction

Instruments that are simpler and faster than Fourier analysers have been developed to yield a finite number of Walsh series coefficients of a signal [1,2]. In many instances the Walsh spectrum of a signal is as meaningful as the Fourier spectrum, and sometimes it is preferable; nevertheless, because of bandwidth restrictions of transmission channels, the Fourier spectrum corresponding to a given Walsh spectrum would often be required.

Given the Walsh coefficients of a signal, the corresponding Fourier coefficients may be evaluated by either a general-purpose or a special-purpose computer, using established conversion formulae [3]. In practical conversion systems, two forms of truncation error may arise. First, the word lengths in the system hardware may be inadequate. Such roundoff errors are here considered to be negligible. Since, in general, the conversion equation for each Fourier coefficient is an infinite sum of products of constants and the given Walsh coefficients, a second and more important source of error is truncation of the infinite series since the number of Walsh coefficients will always be finite.

- Signals fall into four spectral categories;
1. Infinite Walsh series with infinite Fourier series,
 2. Finite Walsh series with finite Fourier series,
 3. Finite Walsh series with infinite Fourier series,
 4. Infinite Walsh series with finite Fourier series.

For instrumentation purposes, the last category is of particular interest. It is shown below that if a band-limited signal with a highest normalized frequency component (harmonic) N is applied to a Walsh analyser whose highest normalized frequency component readout is M , then all N Fourier harmonics of the signal can be determined without error,

provided that $M \geq N$. Thus, instruments that yield a finite number of Walsh coefficients can be used for the precise evaluation of the Fourier coefficients of band-limited signals. Furthermore, a substantial easing of hardware requirements in a special-purpose Walsh to Fourier series converter (or of software requirements in a general-purpose computer conversion) is achieved if one puts $M = 2^x$, where x is an integer. Thus, in the design of Walsh spectral analysers for use on frequency-limited input signals, the highest frequency component to be measured is best chosen to be a Rademacher function of order x . A dual relationship holds for signals in category 3.

Series Conversion

Let a function $f(\theta)$ be represented by a sequence-ordered Walsh series;

$$f(\theta) = A_0 + \sum_{m=1}^{\infty} [A_m \text{cal}(m, \theta) + B_m \text{sal}(m, \theta)] \quad (1)$$

The coefficients A_m and B_m of the even and odd Walsh functions, respectively, are defined by

$$A_m = \int_0^1 f(\theta) \text{cal}(m, \theta) d\theta \quad (2)$$

$$B_m = \int_0^1 f(\theta) \text{sal}(m, \theta) d\theta \quad (3)$$

where m is the normalized frequency and θ is the normalized time. The same function $f(\theta)$ has the corresponding Fourier series

$$f(\theta) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cos 2\pi n\theta + b_n \sin 2\pi n\theta] \quad (4)$$

where

$$a_n = 2 \int_0^1 f(\theta) \cos 2\pi n\theta d\theta \quad (5)$$

$$b_n = 2 \int_0^1 f(\theta) \sin 2\pi n\theta d\theta \quad (6)$$

and where n is the normalized frequency or harmonic number. It is desired to use the Walsh coefficients A_m and B_m in order to derive a_n and b_n . We first consider signals in category 1, i.e., that have both infinite Walsh spectra and infinite Fourier spectra.

The even terms a_n of the Fourier series of a signal are functions only of the even terms A_m of the corresponding Walsh series. Similarly b_n terms depend only on B_m terms. Only the even terms are considered below; similar derivations apply for the odd terms.

The Walsh to Fourier series conversion relation is derived by equating the terms of each series:

$$\sum_{n=1}^{\infty} a_n \cos 2\pi n\theta = \sum_{m=1}^{\infty} A_m \text{cal}(m, \theta) \quad (7)$$

The cal functions are expanded into sets of equivalent Fourier series expressions whose terms have coefficients $a_{n,m}$, where

$$a_{n,m} = 2 \int_0^1 \text{cal}(m, \theta) \cos 2\pi n\theta \, d\theta \quad (8)$$

is the n th Fourier coefficient of $\text{cal}(m, \theta)$. The $m \times n$ matrix of the set $\{a_{n,m}\}$ is denoted F^T . In the expansion of the right-hand side of Eq. (7), terms containing $\cos 2\pi n\theta$ are grouped, yielding a_n values given by

$$a_n = \sum_{m=1}^{\infty} a_{n,m} A_m \quad (9)$$

If a represents the $n \times 1$ matrix of the set $\{a_n\}$ as $n \rightarrow \infty$, and A represents the $m \times 1$ matrix of the set $\{A_m\}$ as $m \rightarrow \infty$, then

$$a = F^T A \quad (10)$$

However, if only a finite number M of Walsh coefficients are known, then a_n can only be approximated as \hat{a}_n , where

$$\hat{a}_n = \sum_{m=1}^M a_{n,m} A_m \quad (11)$$

The coefficients \hat{a}_n can be considered as the Fourier coefficients of a sequence-limited function. The mean-squared error introduced by the approximation is

$$\|a - \hat{a}\|^2 = \left\| \sum_{m=M+1}^{\infty} a_{n,m} A_m \right\|^2 \quad (12)$$

Thus approximation errors in the conversion are dependent on the Walsh series coefficients of the signals. As M increases, errors tend to decrease, but not necessarily in a uniform manner.

Since a number of the constants $a_{n,m}$ are zero, special cases arise for functions with infinite Walsh spectra and infinite Fourier spectra where there is no error due to truncation of Eq. (9), provided that $m > n$. One such case is a sawtooth wave which is periodic over the interval $(0,1)$; it has Walsh series coefficients that are non-zero only for $m=2^x$ where x is an integer, so the conversion equation for each a_n has only one non-zero term.

Sequence-Limited Functions

Signals with spectra in categories 2 and 3 are sequence-limited, with Walsh to Fourier series conversion equations of the form of Eq. (9). If $A_m = 0$ for $m > M$, the mean-squared conversion error term of Eq. (12) is zero provided that at least M Walsh coefficients are used to evaluate a_n .

Frequency-Limited Functions

Functions, other than constants, which have finite Fourier spectra, necessarily have infinite Walsh spectra. If a function $f(\theta)$ is band-limited to become $\hat{f}(\theta)$ with a limited number of harmonics N , and if $\hat{f}(\theta)$ is applied to a Walsh spectrum analyzer (see Fig. 1) to yield the first M Walsh coefficients, then the N Fourier coefficients of $f(\theta)$ can be determined precisely from the M measured Walsh coefficients, provided that $M \geq N$.

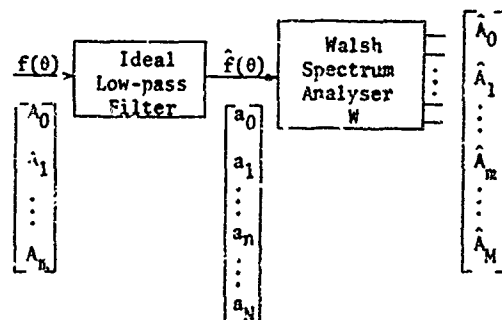


Figure 1. Walsh spectrum analysis of frequency-limited function.

Writing

$$\hat{f}(\theta) = \sum_{n=1}^N a_n \cos 2\pi n\theta \quad (13)$$

then

$$\hat{A}_m = \int_0^1 \hat{f}(\theta) \text{cal}(m, \theta) \, d\theta \quad (14)$$

$$= \sum_{n=1}^N \left[\int_0^1 \cos 2\pi n\theta \text{cal}(m, \theta) \, d\theta \right] a_n$$

or

$$\hat{A}_m = \sum_{n=1}^N A_{m,n} a_n, \quad a_n = 0 \text{ for } n > N \quad (15)$$

where $A_{m,n}$ is the m th Walsh coefficient of $\cos 2\pi n\theta$. In matrix form Eq. (15) is written

$$\hat{A} = N a \quad (16)$$

Since

$$A_{m,n} = \frac{1}{2} a_{n,m} \quad (17)$$

$$N = \frac{1}{2} F^T \quad (18)$$

One can solve for a in terms of \hat{A} as follows:

$$\hat{A} = \frac{1}{2} F^T a \quad (19)$$

$$\hat{A} = \frac{1}{2} F F^T a = K a \quad \text{say,} \quad (20)$$

so that

$$a = K^{-1} F \hat{A} \quad (21)$$

if K is non-singular. The set $\{a_n\}$ can be solved by a system of N linear independent equations. Thus, M must equal or exceed N . It can be shown that K is indeed non-singular for $M \geq N$. Thus, the first N Fourier coefficients can be recovered with no truncation error. As $M \rightarrow \infty$, K becomes the identity matrix and Eq. (21) reduces to Eq. (10).

Dual Relationship

A dual relation permits the determination of Walsh coefficients in terms of Fourier coefficients. Eq. (19) can be used to find the Walsh coefficients of a band-limited function. The M Walsh coefficients of a frequency-limited function are derived from the first N Fourier harmonics, provided that $M \geq N$. From Eq. (10)

$$F^T a = F^T F \hat{A} \quad (22)$$

$$\therefore A = [F^T F]^{-1} F^T a \quad (23)$$

if $[F^T F]$ is non-singular. For $N=M$; i.e., for square matrix F that can be inverted,

$$A = \frac{1}{2} F^T K^{-1} a \quad (24)$$

It is seen that the above equation is similar to Eq. (19) for the band-limited case. As $M \rightarrow \infty$, K^{-1} becomes an identity matrix.

Instrumentation

Digital hardware requirements for a special-purpose Fourier to Walsh or Walsh to Fourier converter (or software requirements for computers to achieve these ends) are eased by using an important property of the matrix K . It can be shown that for $M=2^x$, where x is an integer, K diagonalizes with diagonal elements

$$K_{n,n} = \frac{1}{2} \sum_{m=1}^{2^x} (a_{n,m})^2 \quad (25)$$

$$= \begin{cases} \text{sinc}^2(n/2^{x+1}), & n \leq 2^x \\ 2\text{sinc}^2(n/2^{x+1}), & n > 2^x \end{cases}$$

This property can be established by showing that the rows of F are mutually orthogonal if $M = 2^x \geq N$.

The matrices F and K recur in each of the conversion equations (10), (19), (21) and (24) for band-limited or frequency-limited functions. By taking advantage of the diagonalizing property of K , a minimum set of constants can be stored in read-only memories (ROM) of a digital converter if $M=N=2^x$. Only the non-zero elements

of a $[2^x \times 2^x]$ matrix F and the 2^x diagonal elements of K need be stored. Peripherals about the ROM's are used to program each of the conversion equations. Thus, one digital processor can perform the Walsh to Fourier, or the Fourier to Walsh series conversion.

Conclusion

As a general procedure for Walsh series and Fourier series analysis using a Walsh spectrum analyzer, the following procedure is adopted. If the first N Fourier coefficients of a signal are to be evaluated,

- The signal is passed through a low-pass filter to obtain a function with 2^x or fewer harmonics, i.e., the set a .
- The function is analyzed using a Walsh spectrum analyzer to obtain the first 2^x Walsh coefficients, i.e., the set \hat{A} .
- The set \hat{A} is used in a converter that is programmed to solve Eq. (21) in order to evaluate a . If only the Walsh coefficients of the original signal are to be measured, the lowpass filter is by-passed.

Similarly, a Fourier spectrum analyzer and series converter can be used for precise evaluation of the Walsh coefficients, provided the signal is frequency filtered before analysis.

Acknowledgements

The authors wish to thank R. DeBuda and E. Mosca for many useful suggestions.

References

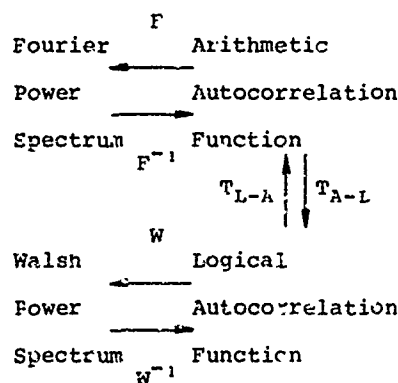
- C.A. Bass (Editor), Proc. 1970 Symposium and Workshop on Applications of Walsh Functions, Washington, D.C., April 1970.
- R.W. Zeek and A.E. Showalter (Editors), Proc. 1971 Symposium on Applications of Walsh Functions, Special Issue, IEEE Trans. Electromagnetic Compatibility, Vol. EMC-13, August 1971.
- K.H. Siemens, Digital Walsh-Fourier Analyser for Periodic Waveforms, M.Eng. Thesis, McMaster University, Hamilton, Ontario, Canada, 1969.

DISCRETE WALSH AND FOURIER POWER SPECTRA

Guner S. Robinson
Member of the Technical Staff
COMSAT Laboratories
Clarksbury, Maryland

Introduction

The Walsh power spectrum of a sequence of random samples is defined as the Walsh or sequency-ordered Hadamard transform of the logical autocorrelation function of the random sequence. The "logical" autocorrelation function is defined in a form similar to the "arithmetic" autocorrelation function. The Fourier power spectrum, which is defined as the Fourier transform of the arithmetic autocorrelation function, can be obtained from the Walsh power spectrum by a linear transformation. The chain of transformations can be summarized as



The top half of this chain is known as the Wiener-Khinchine theorem, while the bottom half is known as the "logical" Wiener-Khinchine theorem. For a given process with computed or modeled autocorrelation function, the Fourier and Walsh power spectra are computed by using fast Fourier and Walsh transforms, respectively. Examples are given from speech and imagery data.

Logical Convolution*

Consider the following random sequence of length $N = 2^n$, where n is an integer:

$$x(j), j = 0, 1, 2, \dots, N-1$$

Such a sequence can be assumed to represent a window (or a block of N samples)

*Also called dyadic convolution.

of a discrete random process. The logical convolution (or correlation*) function based on the m th window is defined as^{1,2}

$$L^{(m)}(k) = \frac{1}{N} \sum_{j=0}^{N-1} x(j \oplus k) x(j) \quad (1)$$

$$k = 0, 1, \dots, N-1$$

where $j \oplus k$ denotes the modulo 2 sum of the integers j and k .

In Equation (1) j and k are non-negative integers less than $2^n - 1$. If

$$G_n = \{0, 1, 2, \dots, 2^n - 1\}$$

represents the set of such integers, then every element of this set can be expressed in binary notation with n coefficients which are either zero or one. Let

$$j = \sum_{i=0}^{n-1} j_i 2^i$$

$$k = \sum_{i=0}^{n-1} k_i 2^i$$

Then the operation \oplus in Equation (1) means

$$j \oplus k = \sum_{i=0}^{n-1} (j_i \oplus k_i) 2^i \quad (2)$$

where $j_i \oplus k_i$ represents the modulo 2 addition (i.e., binary addition without carry) operation. That is,

$$0 \oplus 0 = 1 \oplus 1 = 0$$

$$0 \oplus 1 = 1 \oplus 0 = 1$$

Table I shows the modulo 2 addition matrix for the set $G_n = \{0, 1, \dots, 7\}$, where j indicates the columns and k indicates the rows of the table. The

*Dyadic convolution and correlation are identical because modulo 2 addition and subtraction are identical operations.

first row and column are all of the integers in the set G , in increasing order, since the modulo 2 sum of an integer and zero does not alter the integer. It should be noted that, for a certain n , if the matrix is divided into four $2^{n/2}$ by $2^{n/2}$ matrices, then each matrix is symmetrical with respect to both diagonals. Thus Table I can be generalized and extended for $n > 3$ by inspection.

Table I. Modulo 2 addition, given by $j \oplus k$, for integers j and k between 0 and 7

$j \backslash k$	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	0	3	2	5	4	7	6
2	2	3	0	1	6	7	4	5
3	3	2	1	0	7	6	5	4
4	4	5	6	7	0	1	2	3
5	5	4	7	6	1	0	3	2
6	6	7	4	5	2	3	0	1
7	7	6	5	4	3	2	1	0

Table II, which is derived from Table I, shows that computation of the dyadic autocorrelation function requires varying amounts of time shifts for various k . Time shift k^* , defined by the relationship

$$j + k^* = j \oplus k \quad (3)$$

Table II. Time shift k^* , given by $k^* = (j \oplus k) - j$, for integers j and k between 0 and 7

$j \backslash k$	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	1	-1	1	-1	1	-1	1	-1
2	2	2	-2	-2	2	2	-2	-2
3	3	1	-1	-3	3	1	-1	-3
4	4	4	4	4	-4	-4	-4	-4
5	5	3	5	3	-3	-5	-3	-5
6	6	6	2	2	-2	-2	-6	-6
7	7	5	3	1	-1	-3	-5	-7

has a certain structure; i.e., for any n , if the matrix of Table II is divided into two 2^{n-1} by 2^n matrices, then the elements of these matrices have odd symmetry properties for any k . This is equivalent to saying that, for any n , the right half of the shift matrix is the mirror image of the left half. The particular symmetry properties of the shift matrix are the basis of the recursive relationship between the logical and arithmetic autocorrelation functions which will be derived next.

Logical Autocorrelation Function

The logical autocorrelation function is defined as the expected value of the local logical autocorrelation function of Equation (1):

$$L(k) = E\{L^{(m)}(k)\} \quad (4)$$

where the expectation operator E denotes the ensemble average of M logical autocorrelation functions; i.e.,

$$L(k) = \frac{1}{M} \sum_{m=1}^M L^{(m)}(k) \quad (5)$$

$$k = 0, 1, \dots, N-1$$

The relationship between $L(k)$ and the "arithmetic" autocorrelation function $R(k)$ will be established next. For a stationary process the arithmetic autocorrelation function is even and a function of the time difference only:

$$E\{x(j+k^*)x(j)\} = R(k^*) \quad (6)$$

$$R(-k^*) = R(k^*)$$

For a logical autocorrelation function, if Equation (1) is written in matrix form using the indices of Table I, one obtains a linear combination of $R(k)$ for each $L(k)$. This relationship can be obtained recursively for any n .

The $R(k) \rightarrow L(k)$ transformation may be easily obtained by examining Table II. The first row, which corresponds to $L(0)$, yields the correlation of N samples for zero time shift; thus Equation (1) yields $L(0) = R(0)$. The second row, which corresponds to $L(1)$, yields the expected value of $N/2$ samples shifted one element to the right and $N/2$ samples shifted one element to the left. Since $R(-1) = R(1)$, the second row results in $L(1) = R(1)$. The N th row gives all of the odd shifts up to $N-1$, both to the right and to the left. The results are shown in Table III, where L_k

and R_k denote $L(k)$ and $R(k)$, respectively. It should be noted that

$$|L_k/L_0| \leq 1$$

which is similar to

$$|R_k/R_0| \leq 1$$

Both logical and arithmetic autocorrelation functions can be normalized by R_0 , since $R_0 = L_0 = \sigma^2$ is the variance of the random signal. That is,

$$\ell_k = L_k/\sigma^2 \quad (7a)$$

$$\rho_k = R_k/\sigma^2 \quad (7b)$$

Table III. The relationship between logical and arithmetic autocorrelation functions for time shifts between 0 and 15

Arithmetic to Logical	Logical to Arithmetic
$L_0 = R_0$	$R_0 = L_0$
$L_1 = R_1$	$R_1 = L_1$
$L_2 = R_2$	$R_2 = L_2$
$L_3 = (R_1 + R_7)/2$	$R_3 = 2L_3 - L_1$
$L_4 = R_4$	$R_4 = L_4$
$L_5 = (R_3 + R_5)/2$	$R_5 = 2L_5 - 2L_3 + L_1$
$L_6 = (R_2 + R_6)/2$	$R_6 = 2L_6 - L_2$
$L_7 = (R_1 + R_3 + R_5 + R_7)/4$	$R_7 = 4L_7 - 2L_5 - L_1$
$L_8 = R_8$	$R_8 = L_8$
$L_9 = (R_7 + R_9)/2$	$R_9 = 2L_9 - 4L_7 + 2L_5 + L_1$
$L_{10} = (R_6 + R_{10})/2$	$R_{10} = 2L_{10} - 2L_6 + L_2$
$L_{11} = (R_5 + R_7 + R_9 + R_{11})/4$	$R_{11} = 4L_{11} - 2L_9 - 2L_5 + 2L_3 - L_1$
$L_{12} = (R_4 + R_{12})/2$	$R_{12} = 2L_{12} - L_4$
$L_{13} = (R_3 + R_5 + R_{11} + R_{13})/4$	$R_{13} = 4L_{13} - 4L_{11} + 2L_9 - 2L_5 + L_1$
$L_{14} = (R_2 + R_6 + R_{10} + R_{14})/4$	$R_{14} = 4L_{14} - 2L_{10} - L_2$
$L_{15} = (R_1 + R_3 + R_5 + R_7 + R_9 + R_{11} + R_{13} + R_{15})/8$	$R_{15} = 8L_{15} - 4L_{13} - L_9 - L_1$

Logical Covariance Matrix

Logical and arithmetic autocovariance matrices are formed as

$$\{C_L\}_{j,k} = \sigma^2 \ell(j \oplus k) \quad (8a)$$

$$\{C_A\}_{j,k} = \sigma^2 \rho(j \sim k) \quad (8b)$$

For example, for $n = 2$,

$$C_L = \sigma^2 \begin{bmatrix} 1 & \ell_1 & \ell_7 & \ell_3 \\ \ell_1 & 1 & \ell_3 & \ell_2 \\ \ell_2 & \ell_3 & 1 & \ell_4 \\ \ell_3 & \ell_2 & \ell_4 & 1 \end{bmatrix} \quad (9a)$$

$$C_A = \sigma^2 \begin{bmatrix} 1 & \rho_1 & \rho_2 & \rho_3 \\ \rho_1 & 1 & \rho_1 & \rho_2 \\ \rho_2 & \rho_1 & 1 & \rho_1 \\ \rho_3 & \rho_2 & \rho_1 & 1 \end{bmatrix} \quad (9b)$$

where C_L is a dyadic matrix and C_A is a Toeplitz matrix. The discrete Fourier transform matrix F , whose elements are defined as

$$\{F\}_{j,k} = \frac{1}{N} e^{-i(2\pi/N)jk} \quad (10)$$

$$j, k = 0, 1, \dots, N-1$$

$$i = \sqrt{-1}$$

does not diagonalize either of these matrices.* The dyadic matrix of Equation (9a) can be diagonalized by a Walsh transform matrix W , whose rows are the normalized discrete Walsh functions in order of increasing number of sign changes along each row. For example, for $N = 4$,

$$W = \frac{1}{\sqrt{4}} \begin{bmatrix} + & + & + & + \\ + & + & - & - \\ + & - & - & + \\ + & - & - & + \end{bmatrix}$$

where \pm denotes ± 1 .

*An N by N discrete Fourier transform matrix diagonalizes a circulant matrix which is defined as

$$\{C_c\}_{j,k} = P[(k-j) \bmod N]$$

i.e., each row is obtained by a circular right shift of the preceding row, and thus the elements of the first row are sufficient to determine the matrix. For example, for $N = 4$,

$$C_c = \sigma^2 \begin{bmatrix} 1 & P_1 & P_2 & P_3 \\ P_3 & 1 & P_1 & P_2 \\ P_2 & P_3 & 1 & P_1 \\ P_1 & P_2 & P_3 & 1 \end{bmatrix}$$

$$\rho_i = \frac{P_i}{P_0} = \frac{P_i}{\sigma^2}$$

$$i = 0, 1, 2, 3$$

Diagonalization of the Covariance Matrices

Let the matrix equation

$$Y = TX \quad (11)$$

represent a linear orthonormal transform of the input random process X . The covariance matrix of the output vector process is, by definition,

$$C_Y = E \{ (Y - \bar{Y}) (Y - \bar{Y})^* \} \quad (12)$$

where E is the expectation operator, \bar{Y} is the mean value of Y , and the superscript $*$ denotes the conjugate transpose.

Equations (11) and (12) result in

$$C_Y = TC_X T^* \quad (13)$$

where C_X is the covariance matrix of the input vector. If T is a Walsh transform matrix, Equation (13) yields

$$C_Y = WC_L W \quad (14a)$$

since $W^* = W$. The matrix C_Y is diagonal and the diagonal elements are the Walsh transform coefficients of the vector $\{L_0, L_1, \dots, L_{N-1}\}'$. (The prime denotes the transpose.) For $N = 4$ it can be checked that the elements of the diagonal matrix C_Y are given as

$$\lambda_0^W = \sigma^2 \{1 + L_1 + L_2 + L_3\}$$

$$\lambda_1^W = \sigma^2 \{1 + L_1 - L_2 - L_3\}$$

$$\lambda_2^W = \sigma^2 \{1 - L_1 - L_2 + L_3\}$$

$$\lambda_3^W = \sigma^2 \{1 - L_1 + L_2 - L_3\}$$

If C_X in Equation (13) is the Toeplitz form defined by Equation (9b), the linear transform matrix that diagonalizes C_X is the Karhunen-Loève transform matrix K , whose rows are the eigenvectors of C_X . For a real random process X , K is real but not symmetric. Therefore Equation (13) results in

$$C_Y = KC_X K' \quad (14b)$$

where the prime denotes the transpose. C_Y is diagonal and the diagonal elements are given by the eigenvalues of C_X . For example, for $N = 4$, the eigenvalues of C_X of Equation (9b) have been obtained by using the decomposition approach given by Ray and Driver.³ If $\rho_3 < \rho_2 < \rho_1 < 1$, then the eigenvalues, in decreasing magnitude, are

$$\lambda_0^{KL} = \sigma^2 \left[1 + \frac{\rho_1}{2} + \frac{\rho_3}{2} + \sqrt{\left(\frac{\rho_1}{2} - \frac{\rho_3}{2}\right)^2 + (\rho_1 + \rho_2)^2} \right]$$

$$\lambda_1^{KL} = \sigma^2 \left[1 - \frac{\rho_1}{2} - \frac{\rho_3}{2} + \sqrt{\left(\frac{\rho_1}{2} - \frac{\rho_3}{2}\right)^2 + (\rho_1 - \rho_2)^2} \right]$$

$$\lambda_2^{KL} = \sigma^2 \left[1 + \frac{\rho_1}{2} + \frac{\rho_3}{2} - \sqrt{\left(\frac{\rho_1}{2} - \frac{\rho_3}{2}\right)^2 + (\rho_1 + \rho_2)^2} \right]$$

$$\lambda_3^{KL} = \sigma^2 \left[1 - \frac{\rho_1}{2} - \frac{\rho_3}{2} - \sqrt{\left(\frac{\rho_1}{2} - \frac{\rho_3}{2}\right)^2 + (\rho_1 - \rho_2)^2} \right]$$

In the case of a circulant matrix, Equation (13) results in a diagonal matrix given by

$$C_Y = FC_C F^* \quad (14c)$$

The diagonal elements are the discrete Fourier transform of the vector $\{P_0, P_1, P_2, P_3\}'$. For example, for $N = 4$,

$$\lambda_0^F = \sigma^2 (1 + P_1 + P_2 + P_3)$$

$$\lambda_1^F = \sigma^2 (1 - iP_1 - P_2 + iP_3)$$

$$\lambda_2^F = \sigma^2 (1 - P_1 + P_2 - P_3)$$

$$\lambda_3^F = \sigma^2 (1 + iP_1 - P_2 - iP_3)$$

The Relationship Between Fourier and Walsh Transforms of a Vector

Given an input vector X of size N , the Fourier and Walsh transform coefficient vectors are defined by

$$Y_F = FX$$

$$Y_W = WX$$

Then

$$Y_F = BY_W$$

$$Y_W = B^{-1}Y_F$$

where $B = FW$ is a complex matrix. Since F and W are unitary matrices, B is also unitary; i.e., $B^{-1} = B^*$. The relationship between the Walsh and Fourier

coefficients is a one-to-one linear transformation.

Arithmetic and Logical Wiener-Khintchine Theorems

Let $Y(k)$ be the Walsh transform of the sequence $x(j)$ representing the m th window of the time function. Then the energy spectrum of the window is defined as $S_m(k) = |Y(k)|^2$ for $k = 0, 1, \dots, N-1$. The Walsh power spectrum P_W is defined as the mean of M local energy spectra:

$$P_W(k) = \frac{1}{M} \sum_{m=1}^M S_m(k) \quad (15)$$

Gibbs¹ has shown that $P_W(k)$ is the Walsh transform of $L(j)$ (the logical Wiener-Khintchine theorem); i.e.,

$$P_W(k) \xrightarrow{W} L(j)$$

This transform pair is defined explicitly as

$$P_W(k) = \frac{1}{N} \sum_{j=0}^{N-1} L(j) \text{wal}(k, j, N)$$

$$k = 0, 1, \dots, N-1$$

$$L(j) = \sum_{k=0}^{N-1} P_W(k) \text{wal}(k, j, N)$$

$$j = 0, 1, \dots, N-1$$

where $\text{wal}(k, j, N)$ is the discrete Walsh function of length N defined over the normalized interval from 0 to 1, k is the independent variable taking $N = 2^n$ discrete values, and j is the number of sign changes over the unit interval.

The power spectrum of a random process is the Fourier transform of its autocorrelation function. That is,

$$P_F(k) \xrightarrow{F} R(j)$$

If the linear transformation shown in Table III is denoted as T_{A-L} (transformation from arithmetic to logical), then, for a given random sequence $X(j)$, the power spectra in the Fourier and Walsh domains are related as

$$\begin{array}{ccccc} F(j, k) & \xrightarrow{T_{A-L}} & W(j, k) & & \\ P_F(k) & \xrightarrow{R(j)} & L(j) & \xrightarrow{P_W(k)} & \\ F^{-1}(j, k) & \xleftarrow{T_{L-A}} & W^{-1}(j, k) & & \end{array}$$

where P_F and P_W are the Fourier and Walsh power spectra, and $F(j, k)$ and $W(j, k)$ are the discrete Fourier and Walsh transform matrices, respectively.

This chain of transformations gives the linear relationship between the power spectra in the Fourier and Walsh domains; the left half of this chain is known as the Wiener-Khintchine theorem, while the right half is known as the logical Wiener-Khintchine theorem.

Recursive Relationships Between Logical and Arithmetic Auto-correlation Functions

The transformation given on the left half of Table III can be written in matrix form as⁴

$$T_{A-L} = D_N T_N \quad (16)$$

where D_N is an N by N diagonal matrix whose elements are given by

$$d(j, j) = m_j^{-1}$$

$$m(j) = 2^{V_j - 1 + \delta(j, 0)} \quad (17)$$

$$j = 0, 1, 2, \dots, N-1$$

In Equation (17) V_j represents the number of 1's in the binary representation of j , $\delta(j, 0)$ is the Kronecker delta function, and m_j is shown in Table IV for integers between 0 and 15. In Equation (16), T_N is an N by N matrix which

Table IV. Elements m_j of the inverse diagonal matrix D_N^{-1} for integers between 0 and 15

N (n)	j		V_j	m_j
	Decimal	Binary		
16 (4)	0	0000	0	1
	1	0001	1	1
	2	0010	1	1
	3	0011	2	2
	4	0100	1	1
	5	0101	2	2
	6	0110	2	2
	7	0111	3	4
	8	1000	1	1
	9	1001	2	2
	10	1010	2	2
	11	1011	3	4
	12	1100	2	2
	13	1101	3	4
	14	1110	3	4
	15	1111	4	8

can be generated recursively. (The recursive nature of T_N is essentially outlined by Pichler.⁴ The use of the shuffling matrix introduced in this paper clarifies the recursive algorithm for T_{A-1} and its inverse.) T_N can be generated from $T_{N/2}$ as

$$T_N = \begin{bmatrix} T_{N/2} & 0 \\ T_{N/2} S_{N/2} & T_{N/2} \end{bmatrix} \quad (18)$$

$$N = 1, 2, 4, 8, \dots$$

$$N = 2^n$$

$$n = 0, 1, 2, 3, \dots$$

where the shuffling matrix $S_{N/2}$ is an $N/2$ by $N/2$ matrix whose elements are all zero except those unity elements which are one element off to the right of the SW-NE diagonal. For example,

$$S_1 = 0$$

$$S_2 = \begin{bmatrix} \cdot & \cdot \\ \cdot & 1 \end{bmatrix}$$

$$S_4 = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & 1 & \cdot \\ \cdot & 1 & \cdot & \cdot \end{bmatrix}$$

where each dot denotes a zero.

The inverse of T_N is

$$T_N^{-1} = \begin{bmatrix} T_{N/2}^{-1} & 0 \\ -S_{N/2} T_{N/2}^{-1} & T_{N/2}^{-1} \end{bmatrix} \quad (19)$$

and the recursion starts with $T_1^{-1} = 1$. As an example, T_N and T_N^{-1} are shown next for $N = 16$:

$$\begin{bmatrix} 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & 1 & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & 1 & \cdot & 1 & \cdot & 1 & \cdot & 1 & \cdot & 1 & \cdot & 1 & \cdot & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & -1 & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & -1 & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & -1 & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & -1 & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & -1 & \cdot & \cdot & -1 & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & 1 & \cdot & -1 & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot & -1 & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & -1 & \cdot & 1 & \cdot & -1 & \cdot & \cdot & -1 & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & -1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & -1 & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & -1 & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & -1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -1 & \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\ \cdot & -1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & -1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -1 & \cdot & \cdot & 1 \end{bmatrix}$$

The linear transformation T_{L-A} , which gives the relationship between the logical and arithmetic autocorrelation functions, is the inverse of T_{A-L} of Equation (16). That is,

$$T_{L-A} = T_N^{-1} D_N^{-1} \quad (20)$$

where D_N^{-1} is an N by N diagonal matrix whose elements are given by m_j , $j = 0, 1, \dots, N-1$, of Equation (17).

The simple transformation between the arithmetic and logical autocorrelation functions suggests a more efficient way to compute $R(k)$, $k = 0, 1, \dots, N-1$. Computation of $L(k)$ requires the Walsh transform of the sequence and the Walsh power spectrum of Equation (15) obtained by averaging over overlapping windows. Then a second Walsh transform (the inverse Walsh transform, which is identical to forward Walsh transform) yields $L(k)$.

The fast Walsh transform can also be used for computing the cyclic convolution of two time series.⁵ This approach requires less multiplication than the FFT technique for sequences shorter than $N = 1024$.

Examples

First-Order Markov Process

Experimental evidence indicates that, for most imagery data,^{6,7} the first-order Markov process is a reasonable model for the one-dimensional autocorrelation function. For sampled images, the autocorrelation sequence is

$$R(k) = R(0) \rho^{|k|}$$

$$k = 0, 1, 2, 3, \dots$$

where $R(0) = \sigma^2$ is the variance of the signal sequence. For an image frame of L by L samples (L is typically equal to 512), the biased sample autocovariance function has been computed (for N lags) for each line of the frame

$$R_L(k) = \frac{1}{L} \sum_{i=0}^{L-k} x(i) x(i+k)$$

$$k = 0, 1, \dots, N-1$$

and averaged over the total number of lines. It is seen that, for most

imagery data, there is a high correlation between consecutive samples. A typical value of the correlation coefficient is $\rho = 0.9$.

The discrete Fourier power spectrum is the Fourier transform of the sample autocovariance function multiplied by a suitable weighting function. The weighting function used is the Parzen kernel:

$$h(k) = \begin{cases} 1 - 6 \left(\frac{k}{N-1} \right)^2 + 6 \left| \frac{k}{N-1} \right|^3 & \text{for } |k| < \frac{N-1}{2} \\ 2 \left(1 - \left| \frac{k}{N-1} \right| \right)^3 & \text{for } \frac{N-1}{2} \leq |k| \leq N-1 \\ 0 & \text{for } |k| > N-1 \end{cases}$$

For $N = 32$ and $\rho = 0.9$, $R(k)$, $h(k)$, and their product are shown in Figure 1. This figure also shows the unweighted case for $\rho = 0.5$. Figure 2 shows the logical autocorrelation functions obtained from unweighted and weighted $R(k)$ by using the relationships given in Table III. The discrete values on the curves are joined by straight lines to show several cases on one curve.

Figures 3 and 4 show the Fourier and Walsh power spectra computed from unweighted $R(k)$ and $L(k)$ by means of fast Fourier and fast Walsh transforms. Figures 5 and 6 are the same as Figures 3 and 4, except that N is equal to 16. One should note the similarity of the Fourier and Walsh spectra to the spectral representation of the first-order Markov process in terms of the first 16 eigenvalues of the covariance matrix (Figure 7).

Speech

Continuous speech of 30 seconds duration was sampled at 8000 kHz. The speech material consisted of five test sentences read by a male and female talker.⁸ For $N = 32$ lags, Figure 8 shows the computed autocorrelation function for the male talker, unweighted and weighted by the Parzen kernel. Figure 9 shows the weighted logical autocorrelation function [obtained from $h(k) R(k)$]

and the unweighted logical autocorrelation function for the male talker. Figures 10 and 11 show the Fourier and Walsh power spectra computed from $R(k)$ and $L(k)$ by means of fast Fourier and Walsh transforms. Figures 12 and 13 show the Fourier and Walsh power spectra for $N = 16$ for both talkers.

Figure 14 shows the spectral representation of the speech process (considered to be stationary for a 15-second duration) in terms of the eigenvalues of the covariance matrix. One should note the appearance of a more complex formant structure in the Walsh power spectra (Figures 11 and 13) if no weighting is used; the first "spurious" formant appears at an odd (namely third) multiple of the first formant. Hence, the effect of weighting is to reduce the effect of high-order Walsh coefficients.

Summary

The recursive relationship between arithmetic and logical autocorrelation functions of a stationary process is derived. The Fourier (Walsh) power spectra are computed from the weighted arithmetic (logical) autocorrelation function by means of fast Fourier (Walsh) transforms. Examples from speech and imagery data show that the discrete Fourier and Walsh spectra closely resemble the spectral representation of these processes in terms of eigenvalues and eigenvectors of the covariance matrix.

Acknowledgment

This paper is based upon work performed at COMSAT Laboratories under Corporate sponsorship.

References

1. J. E. Gibbs, "Walsh Spectrometry, a Form of Spectral Analysis Well Suited to Binary Digital Computation," Unpublished Report, National Physical Laboratory, Teddington, Middlesex, England, 1967.
2. J. E. Gibbs and F. R. Pichler, "Comments on Transformation of Fourier Power Spectra into Walsh Power Spectra," Proceedings of the 1971 Symposium on Applications of Walsh Functions, Washington, D.C., April 1971, pp. 51-54.
3. D. W. Ray and R. M. Driver, "Further Decomposition of the Karhunen-Loève Series Representation of a Stationary Random Process," IEEE Transactions on Information Theory, Vol. IT-16, No. 6, November 1970, pp. 663-668.
4. F. R. Pichler, "Some Aspects of a Theory of Correlation with Respect to Walsh Harmonic Analysis," Technical Research Report R-70-11, Department of Electrical Engineering, University of Maryland, College Park, Md., August 1970.
5. D. A. Pitassi, "Fast Convolution Using the Walsh Transform," Proceedings of the 1971 Symposium on the Applications of Walsh Functions, Washington, D.C., April 1971, pp. 130-133.
6. E. R. Kretzmer, "Statistics of Television Signals," Bell System Technical Journal, Vol. 31, July 1952, pp. 761-763.
7. L. E. Franks, "A Model for the Random Video Television Signals," Bell System Technical Journal, Vol. 45, April 1966, pp. 609-630.
8. S. J. Campanella and G. S. Robinson, "A Comparison of Orthogonal Transformations for Digital Speech Processing," IEEE Transactions on Communications Technology, Vol. COM-19, No. 6, Part I, pp. 1045-1050.
9. S. J. Campanella and G. S. Robinson, "Digital Sequence Decomposition of Voice Signals," Proceedings of the 1970 Symposium on Applications of Walsh Functions, Washington, D.C., April 1970, pp. 229-237.

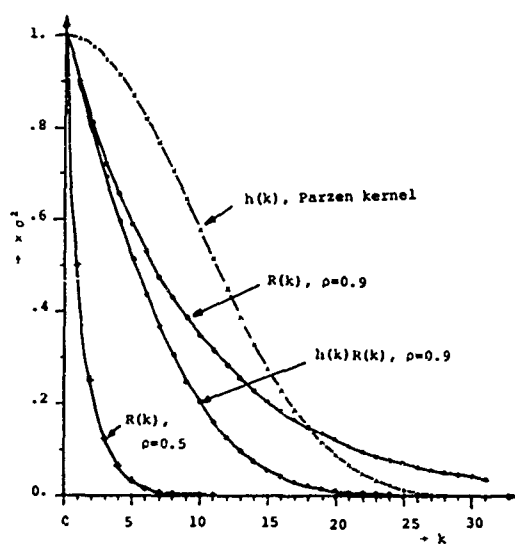


Figure 1. Arithmetic autocorrelation function of the first-order Markov process, $N = 32$

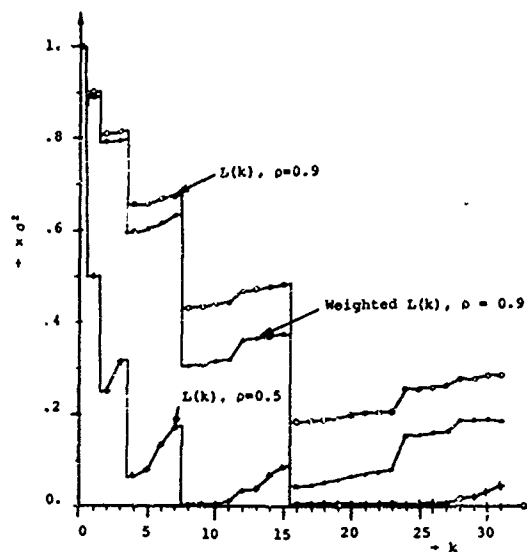


Figure 2. Logical autocorrelation function of the first-order Markov process, $N = 32$

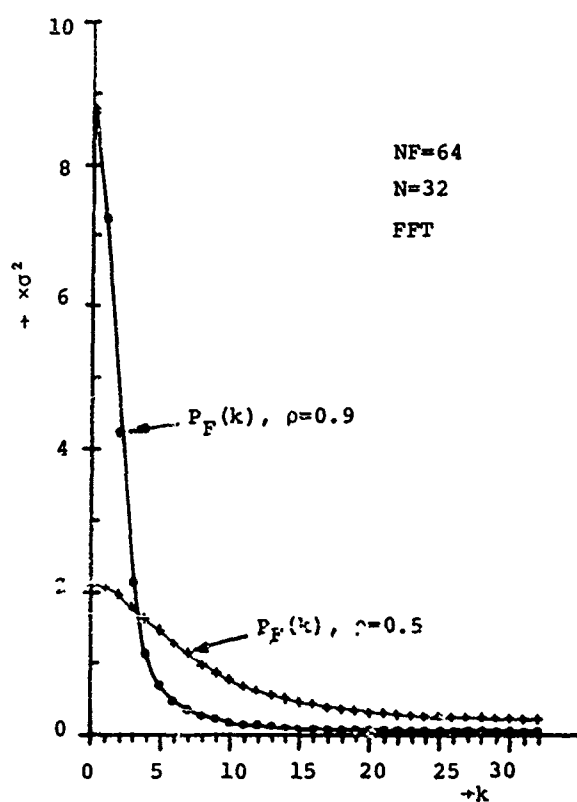


Figure 3. Discrete Fourier power spectrum of the first-order Markov process, $N = 32$

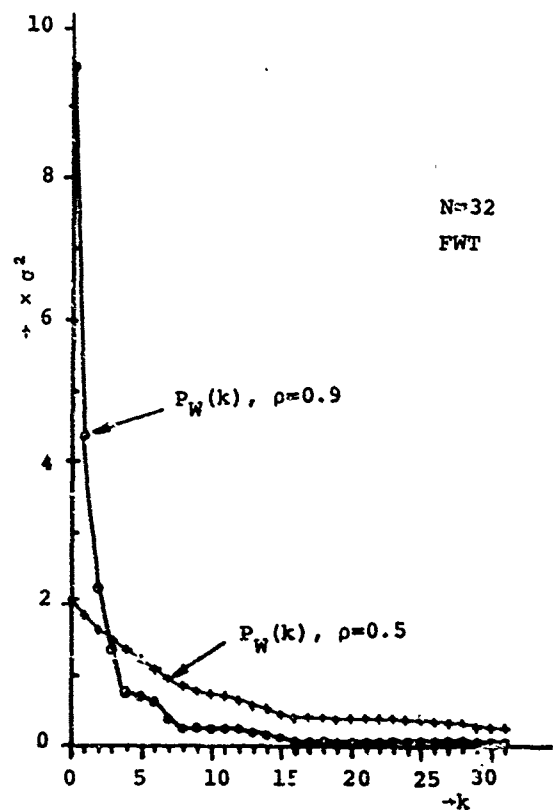


Figure 4. Discrete Walsh power spectrum of the first-order Markov process, $N = 32$

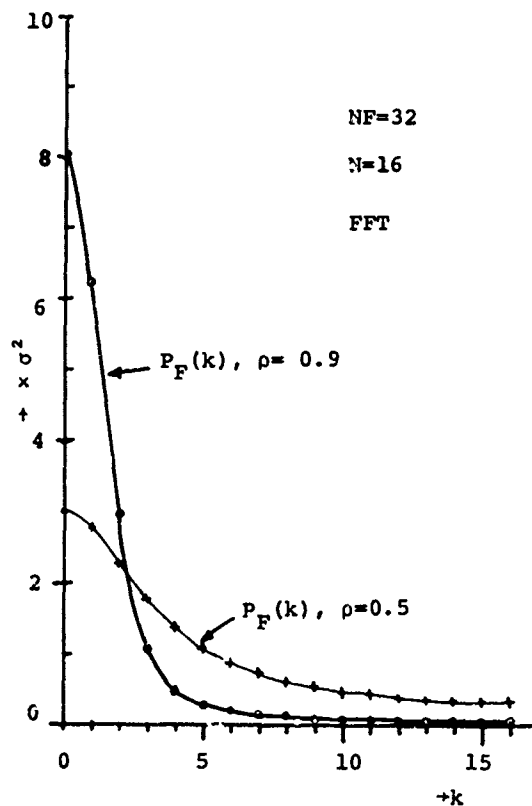


Figure 5. Discrete Fourier power spectrum of the first-order Markov process, $N = 16$

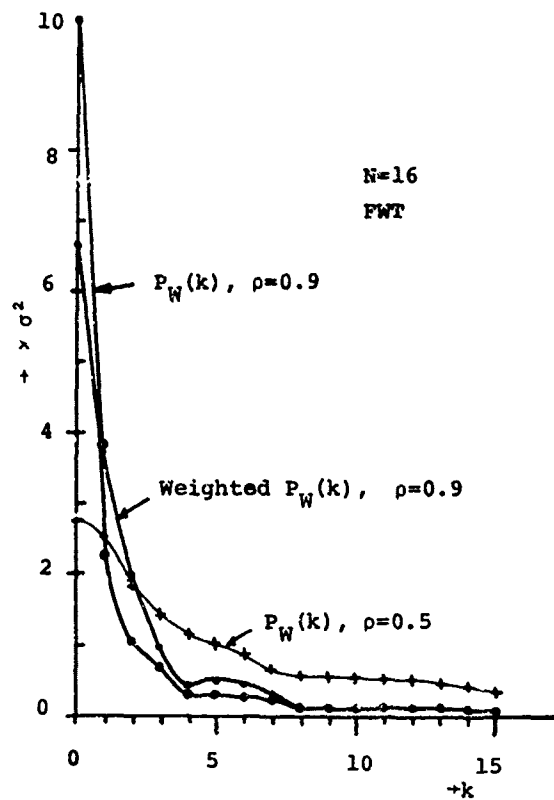


Figure 6. Discrete Walsh power spectrum of the first-order Markov process, $N = 16$

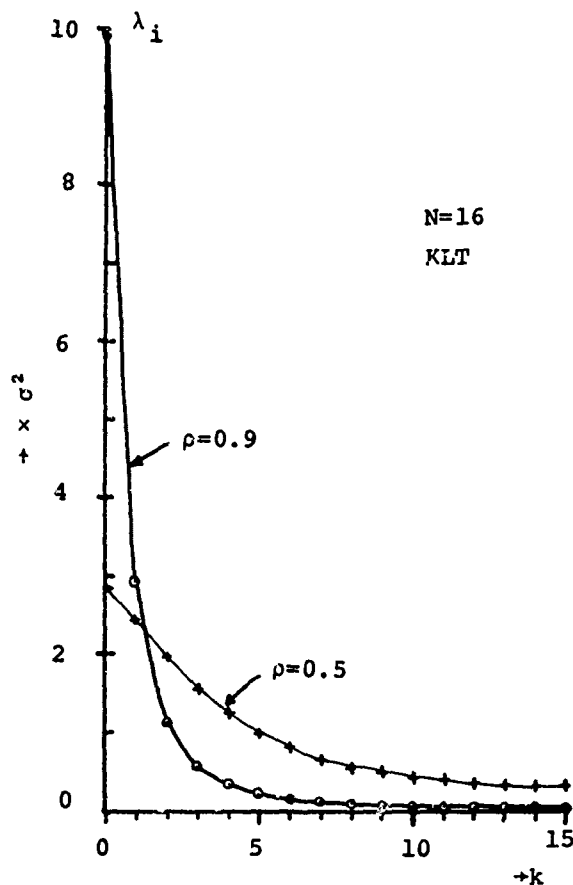


Figure 7. Eigenvalues of the first-order Markov process, $N = 16$

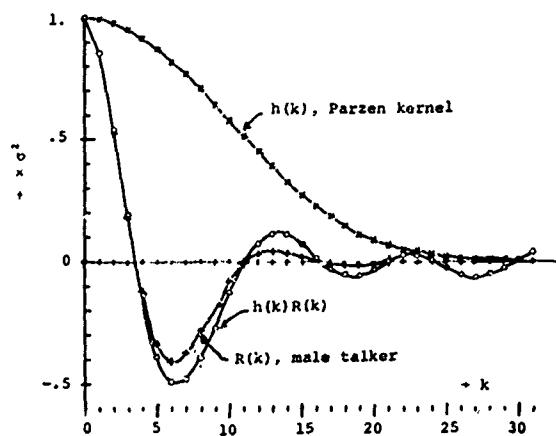


Figure 8. Arithmetic autocorrelation function of speech (male talker), $N = 32$

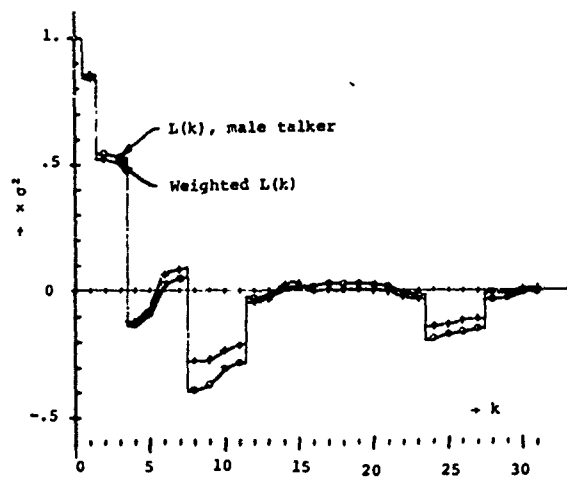


Figure 9. Logical autocorrelation function of speech (male talker), $N = 32$

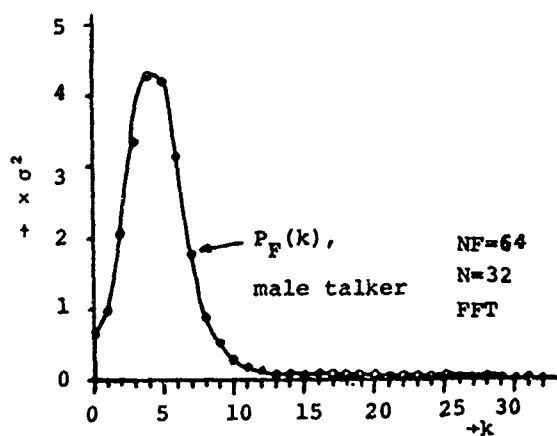


Figure 10. Discrete Fourier power spectrum of speech (male talker), $N = 32$

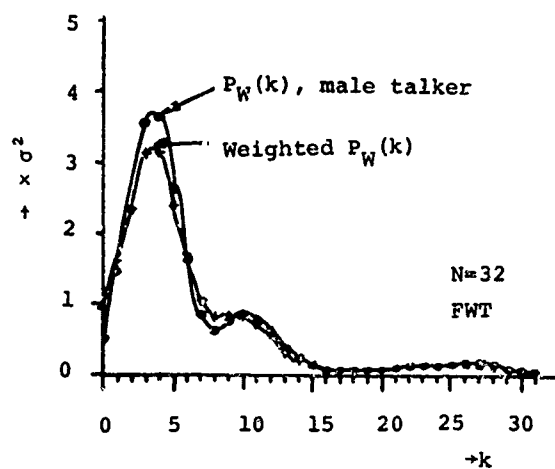


Figure 11. Discrete Walsh power spectrum of speech (male talker), $N = 32$

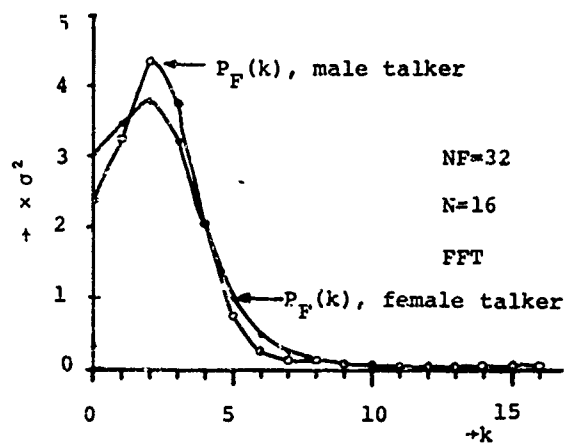


Figure 12. Discrete Fourier power spectrum of speech, $N = 16$

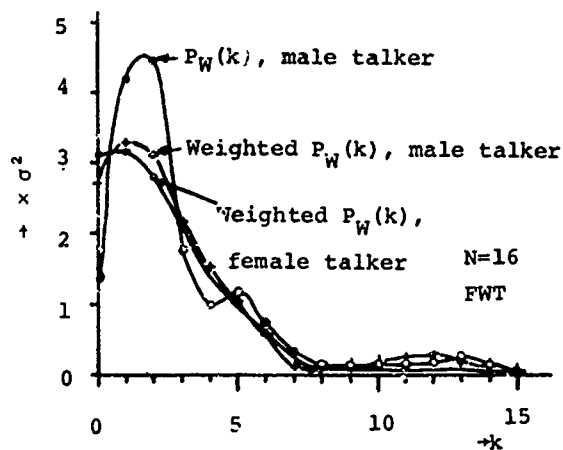


Figure 13. Discrete Walsh power spectrum of speech, $N = 16$

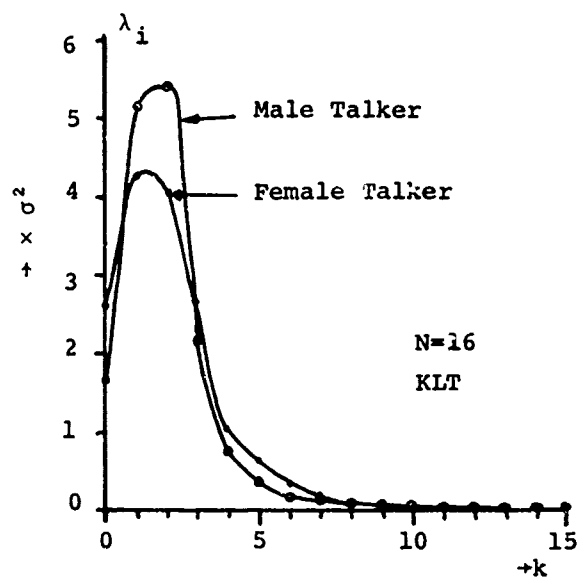


Figure 14. Eigenvalues of the speech covariance matrix for two talkers, $N = 16$

MUTUAL MAPPING OF GENERALIZED CONVOLUTION SYSTEMS

by

H. Gethöffer

Technische Hochschule Darmstadt

Fachbereich Nachrichtentechnik

Darmstadt, West-Germany

Introduction

The theory of linear continuous time systems, the dynamic differential systems, is well known and advanced [1,2,3,4]. The name system theory has been first introduced by Küpfmüller in his famous book on system theory in communication [1]. In recent years the system theory has been generalized to a mathematical concept for abstract relations between input- and output-spaces even for non-physical systems. The map in time-differential systems is given by the Fourier- and Laplace-transform because the frequency domain description is usually clearer than the time domain representation. Modern system analysis is determined in an increasing manner by numerical methods, digital computer simulations and system synthesis by digital hardware circuitries [5,6]. Therefore the theory of generalized linear discrete systems must be advanced. Using suitable generalized sampling theorems the transition from the continuous to the discrete domain will be possible [7]. From the mathematical point of view linear discrete system theory is linear algebra and convenient matrix notations are used in this paper. First a short introduction to discrete generalized convolution theory will be given.

Generalized Convolution

Many discrete linear systems are described by the following input-output-relation

$$y(\nu) = \sum_{k=0}^{2^n-1} h(\nu, k) x(k)$$

with the integers $\nu, k \in N(n) = \{0, 1, 2, \dots, 2^n-1\}$. The equation above represents a generalized convolution system. In the finite linear 2^n -di-

mensional space the standard basis is defined in the usual way

$$\underline{E} = [\underline{e}_0 \ \underline{e}_1 \ \underline{e}_2 \ \dots \ \underline{e}_{2^n-1}]$$

where $\underline{e}_0 \dots \underline{e}_{2^n-1}$ denote the unity vectors. The first unity vector \underline{e}_0 is also called discrete unit impulse δ and this is closely related to the Dirac's function in continuous systems.

$$\underline{e}_0 = \underline{\delta}(0) = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

If the input signal is a unit impulse the output function is called system response or system function. With respect to the definition of the standard basis the discrete signals are written as column vectors

$$\underline{x} = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_k \\ \vdots \\ x_{2^n-1} \end{pmatrix} \quad \underline{y} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_\nu \\ \vdots \\ y_{2^n-1} \end{pmatrix} \quad \underline{h} = \begin{pmatrix} h_0 \\ h_1 \\ h_2 \\ \vdots \\ h_\nu \\ \vdots \\ h_{2^n-1} \end{pmatrix}$$

and the matrix notation of the convolution is given as

$$\underline{y} = \underline{H} \underline{x} = \underline{h} * \underline{x}$$

The matrix \underline{H} is called system matrix and in the special case of convolution the matrix \underline{H} is generated by a general linear operator called translation operator $\underline{Y}^{(k)}$

$$\underline{H} = [\underline{h} | \underline{Y}^{(1)} \underline{h} | \underline{Y}^{(2)} \underline{h} | \dots | \underline{Y}^{(2^n-1)} \underline{h}]$$

$$\underline{H} = [\underline{h} | \underline{h}^{(1)} | \underline{h}^{(2)} | \dots | \underline{h}^{(2^n-1)}]$$

The translation matrix $\underline{V}^{(0)}$ is the unity matrix \underline{E} and the columns of the convolution matrix \underline{H} are given as

$$\underline{h}(k) = \underline{V}^{(k)} \underline{h}$$

If the system matrix \underline{h} is a regular $2^n \times 2^n$ matrix with 2^n linear independent eigenvectors the discrete convolution is mapped by the similarity transform onto the diagonal [8]

$$\underline{G} \underline{H} \underline{G}^{-1} = \text{Diag}(\lambda_i)$$

where λ_i are the eigenvalues of the convolution matrix and \underline{G}^{-1} is called modal-matrix or matrix of the eigenvectors. The transform requires the determination of the eigenvalues and eigenvectors as given by the following equations

$$\det(\underline{H} - \lambda \underline{E}) = 0$$

$$(\underline{H} - \lambda \underline{E}) \underline{G}_i = 0$$

Thus the convolution theorem is defined as

$$\underline{G}(\underline{h} * \underline{x}) = \text{Diag}(\lambda_i) \underline{G} \underline{x}$$

Using the notations of Fichler [9] and defining a componentwise multiplication \bullet the theorem reads as

$$\underline{y} = \underline{h} * \underline{x} \xleftrightarrow{\underline{G}} \underline{\hat{y}} = \underline{\hat{h}} \bullet \underline{\hat{x}}$$

with the following notations

$$\underline{\hat{y}} = \underline{G} \underline{y} \quad \underline{\hat{x}} = \underline{G} \underline{x}$$

$$\underline{\hat{\lambda}} = \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \vdots \\ \lambda_i \\ \vdots \\ \lambda_{2^n-1} \end{bmatrix}$$

and the point-wise multiplication of the vector entries

$$\hat{y}_i = (\hat{\lambda} \bullet \hat{x})_i = \lambda_i \hat{x}_i$$

We call the diagonal matrix of the eigenvalues spectral matrix and the vector of eigenvalues spectral function of the system. A basic block diagram is presented in figure 1.

If we replace the discrete system by a conti-

nuous time-differential system, and the generalized discrete spectral transform \underline{G} by the Laplace transform we get the well known convolution theorem

$$y(t) = h(t) * x(t) \xleftrightarrow{\mathcal{L}} Y(p) = H(p)X(p)$$

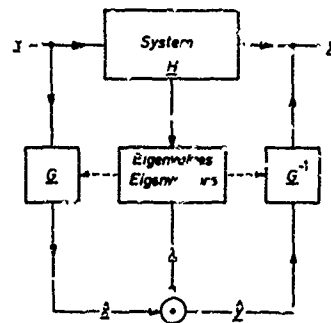


Figure 1. Spectral representation of discrete convolution systems

In these generalized systems the transform \underline{G} is depending on the convolution operator and the system function \underline{h} . In some special types of convolution systems there is a one-to-one correspondence between the eigenvalues and the system function given as

$$\underline{\hat{\lambda}} = \underline{\hat{h}} = \underline{G} \underline{h}$$

and the determination of the eigenvalues and eigenvectors is reduced to a linear transform. The convolution theorem now reads as

$$\underline{y} = \underline{h} * \underline{x} \xleftrightarrow{\underline{G}} \underline{\hat{y}} = \underline{\hat{h}} \bullet \underline{\hat{x}}$$

and the system configuration of figure 1. is simplified to that of figure 2.

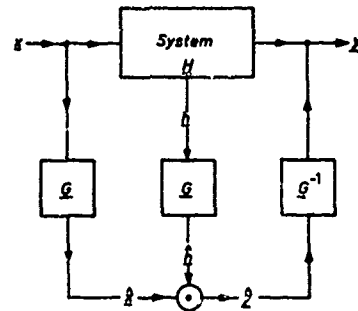


Figure 2. Spectral representation of special convolution systems

For a physical realization task the following theorems must be satisfied

Causality: The system is non-anticipative, if the system matrix is a lower triangular matrix.

with the definition of causality

$$h(\nu, k) = 0 \text{ for } \nu < k$$

the system matrix is given by

$$\underline{H} = \begin{pmatrix} h_{0,0} & & & & \\ h_{1,0} & h_{1,1} & & & \\ h_{2,0} & h_{2,1} & h_{2,2} & & \\ \vdots & \vdots & \vdots & \ddots & \\ h_{2n-1,0} & \dots & \dots & \dots & h_{2n-1,2n-1} \end{pmatrix} \quad \underline{0}$$

Shift invariance: A linear discrete system is shift invariant, iff each shift matrix $\underline{V}^{(k)}$ commutes with the system matrix.

Proof:

$$\begin{aligned} \underline{V}^{(k)} \underline{Y} &= \underline{V}^{(k)} \underline{H} \underline{V}^{(k)-1} \underline{V}^{(k)} \underline{X} \\ &= \underline{H} \underline{V}^{(k)} \underline{X} \\ \underline{H} &= \underline{V}^{(k)} \underline{H} \underline{V}^{(k)-1} \\ \underline{H} \underline{V}^{(k)} &= \underline{V}^{(k)} \underline{H} \\ k &\in \{0, 1, 2, \dots, 2^n - 1\} \end{aligned}$$

Besides this invariant shift matrices must satisfy the following theorem:

The first column of the k-th shift matrix is identical with the k-th unity-vector.

Proof:

$$\begin{aligned} \underline{h}^{(k)} &= \underline{V}^{(k)} \underline{h} = \underline{V}^{(k)} \underline{H} \underline{e}_0 \\ &= \underline{H} \underline{V}^{(k)} \underline{e}_0 = \underline{H} \underline{e}_k \end{aligned}$$

Mutual Mappings

Two arbitrary systems with different matrices are distinguished by the index (a) and (b). Using the matrix notations of the foregoing chapter we write

$$\underline{Y}_{(a)} = \underline{H}_{(a)} \underline{X} \quad \underline{Y}_{(b)} = \underline{H}_{(b)} \underline{X}$$

In both systems the input function must be the same. There exist three basic mutual mappings between the two systems and we define

1) mutual input mapping

$$\underline{X}_{(a)} : \underline{X} \rightarrow \underline{X}_{(a)} \Rightarrow \underline{H}_{(b)} : \underline{X}_{(a)} \rightarrow \underline{Y}_{(b)}$$

$$\underline{X}_{(b)} : \underline{X} \rightarrow \underline{X}_{(b)} \Rightarrow \underline{H}_{(a)} : \underline{X}_{(b)} \rightarrow \underline{Y}_{(a)}$$

2) mutual output mapping

$$\underline{Y}_{(a \rightarrow b)} : \underline{Y}_{(a)} \rightarrow \underline{Y}_{(b)}$$

$$\underline{Y}_{(b \rightarrow a)} : \underline{Y}_{(b)} \rightarrow \underline{Y}_{(a)}$$

3) mutual system mapping

$$\underline{\hat{H}}_{(a \rightarrow b)} : \underline{G}_{(b)} \underline{X} \rightarrow \underline{G}_{(b)} \underline{Y}_{(a)}$$

$$\underline{\hat{H}}_{(b \rightarrow a)} : \underline{G}_{(a)} \underline{X} \rightarrow \underline{G}_{(a)} \underline{Y}_{(b)}$$

Figure 3. is illustrating the three basic maps between the two linear discrete systems.

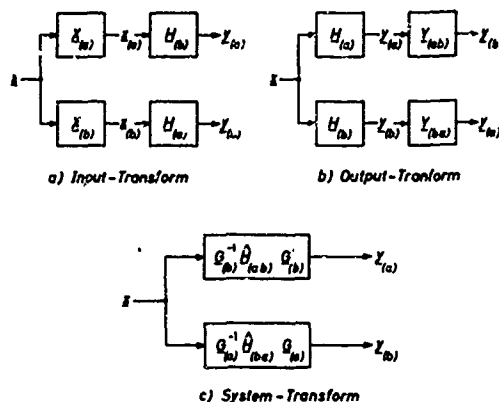


Figure 3. Block diagram of the three basic mappings between two linear systems

Convenient matrix descriptions of the basic mappings are presented as follows:

$$\underline{X}_{(a)} = \underline{X}_{(a)} \underline{X} \quad \underline{X}_{(b)} = \underline{X}_{(b)} \underline{X}$$

The input mapping is done by a linear transformation $\underline{X}_{(a)}$, $\underline{X}_{(b)}$, representing also discrete systems defined as

$$\underline{X}_{(a)} = \underline{H}_{(b)}^{-1} \underline{H}_{(a)}, \quad \underline{X}_{(b)} = \underline{H}_{(a)}^{-1} \underline{H}_{(b)}$$

The two input transform systems are mutually inverse

$$\underline{X}_{(a)} = \underline{X}_{(b)}^{-1}$$

The mutual output transform defines the relations between the output-functions of the discrete systems also representing an output-transform system with the system matrices

$$\underline{Y}_{(b)} = \underline{Y}_{(a \rightarrow b)} \underline{Y}_{(a)} = \underline{H}_{(b)} \underline{H}_{(a)}^{-1} \underline{Y}_{(a)}$$

$$\underline{Y}_{(a)} = \underline{Y}_{(b \rightarrow a)} \underline{Y}_{(b)} = \underline{H}_{(a)} \underline{H}_{(b)}^{-1} \underline{Y}_{(b)}$$

and also the output-transforms are inverse to each other

$$\underline{Y}_{(a \rightarrow b)} = \underline{Y}_{(b \rightarrow a)}^{-1}$$

For a spectral representation of the two systems we use the spectral-output-transform defined as

$$\hat{\underline{Y}}_{(b)} = \hat{\underline{Y}}_{(a \rightarrow b)} \hat{\underline{Y}}_{(a)} = \underline{G}_{(b)} \underline{Y}_{(b)}$$

$$\hat{\underline{Y}}_{(a)} = \hat{\underline{Y}}_{(b \rightarrow a)} \hat{\underline{Y}}_{(b)} = \underline{G}_{(a)} \underline{Y}_{(a)}$$

$$\hat{\underline{Y}}_{(a \rightarrow b)} = \underline{\text{Diag}}(\lambda_{i(b)}) \underline{G}_{(b)} \underline{G}_{(a)}^{-1} \underline{\text{Diag}}\left(\frac{1}{\lambda_{i(a)}}\right)$$

$$\hat{\underline{Y}}_{(b \rightarrow a)} = \underline{\text{Diag}}(\lambda_{i(a)}) \underline{G}_{(a)} \underline{G}_{(b)}^{-1} \underline{\text{Diag}}\left(\frac{1}{\lambda_{i(b)}}\right)$$

and $\hat{\underline{Y}}_{(a)}$ and $\hat{\underline{Y}}_{(b)}$ are the spectral outputs of the systems.

We are now interested in developing matrix notations for the mutual system mapping. The two convolution systems are mapped by the two similarity-transforms onto the diagonal spectral matrices

$$\underline{\text{Diag}}(\lambda_{i(a)}) = \underline{G}_{(a)} \underline{H}_{(a)} \underline{G}_{(a)}^{-1}$$

$$\underline{\text{Diag}}(\lambda_{i(b)}) = \underline{G}_{(b)} \underline{H}_{(b)} \underline{G}_{(b)}^{-1}$$

If the two considered systems are different the following transform yields no diagonal spectral matrices

$$\hat{\underline{H}}_{(a \rightarrow b)} = \underline{G}_{(b)} \underline{H}_{(a)} \underline{G}_{(b)}^{-1}$$

$$\hat{\underline{H}}_{(b \rightarrow a)} = \underline{G}_{(a)} \underline{H}_{(b)} \underline{G}_{(a)}^{-1}$$

The convolution theorem now reads as

$$\underline{G}_{(b)} (\underline{H}_{(a)} * \underline{X})_{(a)} = \hat{\underline{H}}_{(a \rightarrow b)} \underline{G}_{(b)} \underline{X}$$

$$\underline{G}_{(a)} (\underline{H}_{(b)} * \underline{X})_{(b)} = \hat{\underline{H}}_{(b \rightarrow a)} \underline{G}_{(a)} \underline{X}$$

The transformation onto the spectral domain of the other system is shown in figure 4. This map will allow a comparison between two arbitrary convolution systems in the spectral domain. By means of mapping high sophisticated convolution systems onto the spectral domain of well known systems as for example onto the discrete frequency or Fourier domain this also will help us to have a better understanding of new convolution operations.

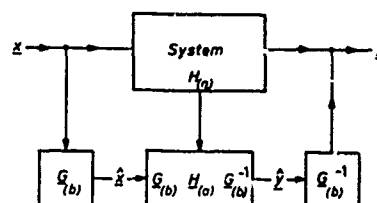


Figure 4. Mutual system mapping

Because calculations operating in the spectral domain with the spectral functions are often easier than the convolution operation in the original domain we now derive matrix notations using the spectral functions

$$\hat{\underline{H}}_{(a \rightarrow b)} = \underline{G}_{(b)} \underline{G}_{(a)}^{-1} \underline{\text{Diag}}(\lambda_{i(a)}) \underline{G}_{(a)} \underline{G}_{(b)}^{-1}$$

$$\hat{\underline{H}}_{(b \rightarrow a)} = \underline{G}_{(a)} \underline{G}_{(b)}^{-1} \underline{\text{Diag}}(\lambda_{i(b)}) \underline{G}_{(b)} \underline{G}_{(a)}^{-1}$$

where

$$\underline{G}_{(b)} \underline{G}_{(a)}^{-1} = \underline{T}_{(a \rightarrow b)}$$

$$\underline{G}_{(a)} \underline{G}_{(b)}^{-1} = \underline{T}_{(b \rightarrow a)}$$

and we call the matrices \underline{T} mutual spectral transform with the following relation

$$\underline{T}_{(a \rightarrow b)} = \underline{T}_{(b \rightarrow a)}^{-1}$$

In the special case of integral transforms in linear continuous systems there were given in [10] three-dimensional representations for the mutual spectral transformation of Walsh- and trigonometric functions.

Cyclic and dyadic systems

In the foregoing chapters we have derived some general matrix notations and now we want to apply this on two systems: circular and dyadic convolution systems.

The circular or cyclic convolution is well known and that because of the strong physical background. The differential-systems are replaced by the difference-systems and the input-output relation is given by

$$y(\nu) = \sum_{k=0}^{2^n-1} h(\nu-k)x(k)$$

$$\nu, k \in N(n) = \{0, 1, 2, \dots, 2^n-1\}$$

The relation to the generalized convolution is given by

$$h(\nu, k) := h(\nu-k)_{\text{mod } 2^n}$$

If we have

$$h(\nu, k) := h(k-\nu)_{\text{mod } 2^n}$$

we call this system a circular correlation system. The shift operator is represented by a unity matrix with permuted columns defined as

$$\underline{V}^{(k)} = [e_{0-k} | e_{1-k} | \dots | e_{\nu-k} | \dots | e_{2^n-1-k}]$$

The convolution matrix is a circular matrix and the cyclic convolution reads as

$$\underline{y} = \underline{h} * \underline{x} = \underline{\text{Cycl}}(\underline{h}_\nu) \underline{x}$$

The similarity transform is given by [11]

$$\underline{F} \underline{\text{Cycl}}(\underline{h}_\nu) \underline{F}^{-1} = \underline{\text{Diag}}(\hat{\underline{h}}_1)$$

\underline{F} is the complex Fourier-matrix defined as

$$\underline{F} = \{ \exp(-j2\pi i k / 2^n) \}$$

The matrix is symmetric and orthogonal

$$\frac{1}{2^n} \underline{F} \underline{F}^* = \underline{E}$$

The cyclic convolution theorem is defined as

$$\underline{F}(\underline{h} * \underline{x}) = \hat{\underline{h}} \bullet \hat{\underline{x}}$$

The basic rules of dyadic convolution theory were first introduced by Pichler [12, 13, 14] and Gibbs [15, 16]. A short introduction shall be given here.

With

$$h(\nu, k) := h(\nu \oplus k)$$

the dyadic convolution is written as a special case of the generalized convolution

$$y(\nu) = \sum_{k=0}^{2^n-1} h(\nu \oplus k) x(k)$$

The dyadic shift operator may be defined as [15]

$$\underline{V}^{(k)} = [e_{0 \oplus k} | e_{1 \oplus k} | \dots | e_{\nu \oplus k} | \dots | e_{2^n-1 \oplus k}]$$

The dyadic convolution is written in matrix notation using the dyadic convolution matrix

$$\underline{y} = \underline{h} \oplus \underline{x} = \underline{\text{Dyad}}(\underline{h}_\nu) \underline{x}$$

because of the commutative operation \oplus the dyadic convolution matrix is symmetric and dyadic convolution and correlation are the same. The dyadic convolution theorem is given as

$$\underline{W}(\underline{h} \oplus \underline{x}) = \underline{h} \bullet \underline{x} = \hat{\underline{y}}$$

with the Walsh-similarity transform

$$\underline{W} \underline{\text{Dyad}}(\underline{h}_\nu) \underline{W}^{-1} = \underline{\text{Diag}}(\hat{\underline{h}}_1)$$

where \underline{W} is the real Walsh matrix and $\hat{\underline{h}}$, $\hat{\underline{y}}$ are the Walsh transforms

$$\underline{W} = \underline{W}^T = 2^n \underline{W}^{-1}$$

$$\hat{\underline{h}} = \underline{W} \underline{h} \quad \hat{\underline{x}} = \underline{W} \underline{x} \quad \hat{\underline{y}} = \underline{W} \underline{y}$$

Dyadic systems have not the same physical interpretation as cyclic systems and this is why not all applications are based on dyadic system theory. But the dyadic system theory is a very young theory and there exist already interesting applications. The main reason therefore is given by the Walsh matrix and its excellent properties. There exist fast transform algorithms reducing the required operations from 2^{2n} to $n2^n$ and in Walsh transformations only additions and subtractions are required.

We are now interested in the mutual system transformation of dyadic convolution onto cyclic Fourier systems. This will help us to have a better understanding of the dyadic operations by means of explaining dyadic systems in the discrete Fourier domain. The mutual system mapping is given by

$$y_{\text{dyad}} = F^{-1} F \underline{\text{Dyad}}(h_\nu) F^{-1} F x$$

with the system transform

$$\hat{H}_{(d \rightarrow c)} = F \underline{\text{Dyad}}(h_\nu) F^{-1}$$

$$\hat{H}_{(d \rightarrow c)} = \frac{1}{2^n} F \underline{\text{Dyad}}(h_\nu) F^{*t}$$

Generally the system transformation is not diagonal because dyadic convolution matrices are only diagonalized by Walsh-transform and not by Fourier-transform. Because of the symmetry of the matrices the mutual system transform is symmetric

$$\hat{H}_{(d \rightarrow c)} = \hat{H}_{(c \rightarrow d)}^t$$

The mutual system transformation $\hat{H}_{(d \rightarrow c)}$ has only 2^n differing elements and

$$(2 \cdot 2^{2n} - 1) / 3 \text{ zero entries.}$$

For $n=2$ the transform is given using the system response entries h_0, h_1, h_2, h_3 by

$$\hat{H}_{(d \rightarrow c)} = \begin{bmatrix} (h_0 + h_1 + h_2 + h_3) & 0 & 0 & 0 \\ 0 & -j/2(h_1 - h_3) & 0 & 1/2(h_0 - h_2) \\ 0 & 0 & (h_0 - h_1 + h_3 - h_2) & 0 \\ 0 & 1/2(h_0 - h_2) & 0 & j/2(h_1 - h_3) \end{bmatrix}$$

The entries of the transform are themselves linear combinations of the system function \underline{h}

and in this special case only two entries outside the diagonal will exist. A block diagram of the dyadic filter in the Fourier domain is presented in figure 5. FFT and iFFT perform the transformation and inverse transformation with fast algorithms. The linear transform and the non-diagonal spectral multiplication scheme represent the proper system transformation. In cyclic convolution the block named linear transform is replaced by the Fourier-transform and the multiplication scheme is completely diagonal. We see now, how dyadic systems are to be described in the Fourier domain: additional linear combinations are required between spectral components.

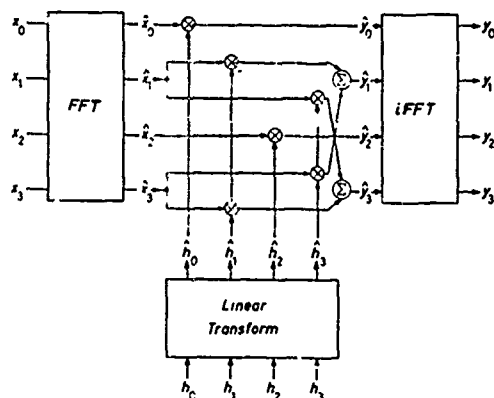


Figure 5. Dyadic convolution in the discrete Fourier domain

Using the Walsh spectral function given by

$$\hat{h}_W = W \underline{h}$$

where \underline{h} is the dyadic system response the mapping of the dyadic convolution system reads

$$\hat{H}_{(d \rightarrow c)} = \frac{1}{2^{2n}} F W \underline{\text{Diag}}(\hat{h}_W) W F^{*t}$$

The transform of the circular convolution onto the Walsh domain is written as

$$y_{\text{cycl}} = \frac{1}{2^{2n}} W W \underline{\text{Cycl}}(h_\nu) W W x$$

with the mutual system mapping

$$\hat{H}_{(c \rightarrow d)} = \frac{1}{2^n} W \underline{\text{Cycl}}(h_\nu) W$$

The transform of a circular matrix yields no diagonal spectral matrix, but only

$$(2^{2n} + 2)/3$$

entries are differing from zero and also 2^n are only different. If the system response is a real function the system transform is also a real matrix and no complex operations are required as in the Fourier mapping. Using fast Walsh transforms FWT the total number of operations consisting only of additions and subtractions is reduced to $n2^n$. An algorithm for this transform has already been given [17] and the so-called cyclic fast Walsh convolution works even faster than the fast Fourier convolution.

For a system with $n=2$ the mutual system transform is written as

$$\hat{H}(c \rightarrow d) = \begin{pmatrix} (h_0, h_1, h_2, h_3) & 0 & 0 & 0 \\ 0 & 1/2(h_0 - h_2) & 1/2(h_1 - h_3) & 0 \\ 0 & -1/2(h_1 \cdot h_3) & 1/2(h_0 - h_2) & 0 \\ 0 & 0 & 0 & (h_0 - h_1 + h_2 - h_3) \end{pmatrix}$$

where h_0, h_1, h_2, h_3 are the elements of the system response.

Fourier convolution systems are often described by the spectral function defined as

$$\hat{h}_F = F \underline{h}$$

The mutual system transform using the spectral representation of the cyclic system is then written as

$$\underline{H}(c \rightarrow d) = \frac{1}{2^{2n}} \underline{W} \underline{F}^{**} \underline{\text{Diag}}(\hat{h}_{Fi}) \underline{F} \underline{W}$$

Figure 6. shows a block diagram of the cyclic convolution mapped onto the Walsh domain. The considered systems are linear and the generalized mapping is bilinear. Therefore the basic rules as stated in the mapping of a dyadic convolution onto the Fourier domain are the same in the other case. The representation of a cyclic convolution system in the Walsh domain is not diagonal and additional linear combinations between spectral components are required. If we replace the linear transform and the non-diagonal multiplication scheme by the Walsh

transform and the diagonal multiplication of the vector components the system turns to a dyadic convolution system.

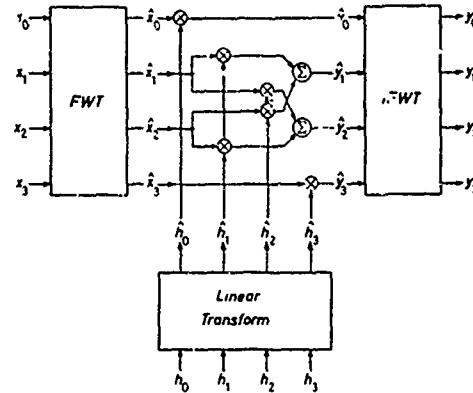


Figure 6. Cyclic convolution in the Walsh domain

Conclusion

Dyadic and cyclic convolution systems are only special systems under a generalized theory. In linear systems there exist close relations between different systems and three basic mappings are presented here. This will help us to understand new systems by well known ones if we map the new systems onto classical domains. The mutual mapping will also give us new methods for the realization of classical tasks in a new system.

Acknowledgement

The author wishes to thank Prof. W. Klein for stimulating this work.

References

- [1] Küpfmüller, K. Die Systemtheorie der Elektrischen Nachrichtenübertragung 3. ed. 1968 Hirzel/ Stuttgart/Germany
- [2] Wunsch, G. Moderne Systemtheorie, Akademische Verlagsgesellschaft-Leipzig 1962
- [3] Unbehauen, R. Systemtheorie, 1969 Oldenburg/München-Wien,
- [4] Kalman, R.E., Falb, P.L., Arbib, M.A., Topics in Mathematical System Theory 1969 McGraw Hill/New York

- [5] Kuo, F.F., Kaiser, J.F., System Analysis by Digital Computer, J. Wiley & Sons / New York 1966
- [6] Gold, B., Rader, C.M., Digital Processing of Signals, McGraw Hill / New York
- [7] Kluvanec, I. Sampling Theorem in Abstract Harmonic Analysis, Mathematicko fyzikalny Casopis. Sloven. Akad. Viec. 15 / 1965 pp. 43-48
- [8] Zurmühl, R., Matrizen, 3.ed. Springer / Berlin 1961
- [9] Pichler, F., On Discrete Dyadic Systems Proc. of the 1971 Symp. on Theory and Applications of Walsh Functions, June 1971 The Hatfield Polytechnic, Hatfield / Hert, U.K.
- [10] Bößwetter, C., Gethöffer, H. Die Gegenseitige Spektraldarstellung von Walsh Funktionen und Trigonometrischen Funktionen, Nachrichtentechn. Zeitschrift NTZ 24/1971 pp. 189-192
- [11] Klein, W. Die Quantisierte Fourier Transformation, Archiv der Elektr. Übertragung AEÜ, 23 (1969) 6 pp. 295-300
- [12] Pichler, F. Walsh Functions and Linear System Theory, Proc. of the 1970 Symp. on Applications of Walsh Functions, Naval Research Lab. Washington D.C. pp. 175-182
- [13] Pichler, F. On State Space Description of Linear Dyadic Invariant Systems, Proc. of the 1971 Symp. on Appl. of Walsh Functions, Trans. Electro. Magn. Comp. EMC-13(1971) 3 pp. 166-170
- [14] Pichler, F., Mathematische Systemtheorie De Gruyter / Berlin to be published
- [15] Gibbs, J.E., Some Properties of Functions on the non-negative Integers less than 2^n , National Physical Lab NPL / DES-Report 3 Teddington, Middlesex U.K. 1969
- [16] Gibbs, J.E., Some Methods of Solution of linear ordinary Logical Differential Equations, DES-Report 2, NPL Teddington Middlesex, England, 1969
- [17] Pitassi, D.A. Fast Convolution Using Walsh Transforms, Proc. of the 1971 Symp. on Appl. of Walsh Functions, EMC-13(1971) pp. 130-133

A CLASS OF EFFICIENT CONVOLUTION ALGORITHMS

Warren F. Davis
Max-Planck-Institut für Radioastronomie
53 Bonn, West Germany

Introduction

Pitassi [1] has described an efficient algorithm for forming the cyclic convolution (or correlation) of sample sequences of length $N = 2^n$. Each iteration of his algorithm requires three sub-convolutions of sequences of length one-half the starting length for that level of iteration. The total number of multiplications is thereby reduced from the classical value of N^2 to $2 \cdot 3^{n-1}$. Successive halvings at each level of iteration are accomplished by parisection [2] of the input and output sequences. This technique is remarkably similar to one of two distinct methods employed in the derivation of the FFT, decimation-in-time [3]. The other FFT method, decimation-in-frequency, involves successive halvings by bisection.

By analogy with the two FFT methods, efficient cyclic (periodic) and linear (aperiodic) convolution algorithms can be found by the methods of parisection and bisection. Thus Pitassi's algorithm is a member of a more general class. All algorithms in the class are characterized by three sub-convolutions of half-length sequences at each level of iteration. Moreover, these algorithms are more efficient than FFT convolution methods when suitably short partial results are required.

Figure 1 is a schematic representation of the general form taken by one iteration of any of the algorithms in the class. Input vector \bar{x} enters plane A, vector \bar{y} plane B, and the output \bar{r} is taken from plane C. The three characteristic sub-convolutions are represented by the boxes a, b, and c at the junction of the A, B, and C-planes.

The complete algorithm for the general case $N = 2^n$ is obtained by iteratively substituting a form similar to Figure 1, but of half-length, into each sub-convolution box. The algorithm is complete when convolutions of length one are finally required. We shall specify, in the derivations which follow, the specific flow which must comprise planes A, B, and C for each technique.

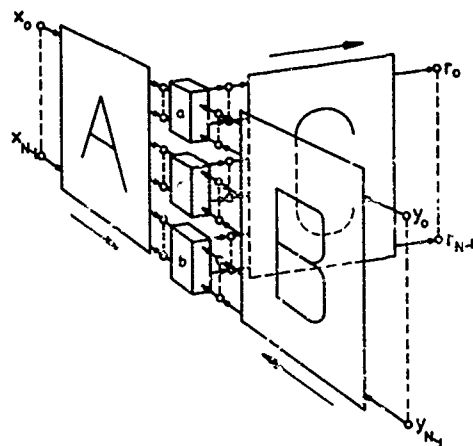


Figure 1. The general form of one iteration of the algorithms in the class. The flow in the A, B, and C-planes is specific to the kind of convolution and the method of derivation.

Operators and Notation

Sequences to be convolved are assumed to be of length $N = 2^n$. They will be represented by column vectors, denoted by overbar, with element indices in the range $0, 1, \dots, N-1$. For example,

$$\bar{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{N-1} \end{bmatrix}$$

The cyclic convolution of two sequences \bar{x} and \bar{y} will be denoted by $\bar{x} \circledast \bar{y}$. If $\bar{r} = \bar{x} \circledast \bar{y}$, then

$$r_k = \sum_{p=0}^{N-1} x_p y_{(p+k) \bmod N} \quad (1)$$

($k=0, 1, \dots, N-1$)

Linear convolution will be denoted by $*$. If $\bar{F} = \bar{X} * \bar{Y}$, then

$$r_k = \sum_{p=0}^{N-1-k} x_p y_{p+k} \quad (2)$$

($k=0,1,\dots,N-1$)

A third convolution, which we shall call the complementary convolution, will be denoted by \boxplus . If $\bar{F} = \bar{X} \boxplus \bar{Y}$, then

$$r_0 \equiv 0,$$

$$r_k = \sum_{p=0}^{k-1} x_{p+N-k} y_p \quad (3)$$

($k=1,2,\dots,N-1$)

Figure 2 illustrates these three convolutions for the case $N = 4$. It is clear from the figure, and can be verified from (1), (2), and (3), that

(a)

y_0	x_0	x_3	x_2	x_1
y_1	x_1	x_0	x_3	x_2
y_2	x_2	x_1	x_0	x_3
y_3	x_3	x_2	x_1	x_0
	r_0	r_1	r_2	r_3

$\bar{F} = \bar{X} \boxplus \bar{Y}$

(b)

y_0	x_0			
y_1	x_1	x_0		
y_2	x_2	x_1	x_0	
y_3	x_3	x_2	x_1	x_0
	r_0	r_1	r_2	r_3

$\bar{F} = \bar{X} * \bar{Y}$

(c)

y_0	0	x_3	x_2	x_1
y_1	0	0	x_3	x_2
y_2	0	0	0	x_3
y_3	0	0	0	0
	r_0	r_1	r_2	r_3

$\bar{F} = \bar{X} \boxminus \bar{Y}$

Figure 2. Cyclic (a), linear (b), and complementary (c) convolutions of \bar{X} and \bar{Y} for $N = 4$. The i -th element of the result \bar{F} is the sum of products between the elements of \bar{Y} and horizontally opposed elements of \bar{X} in the i -th column.

$$\boxplus = * + \boxminus \quad (4)$$

All three convolutions are distributive over addition and are sensitive to the order of factors. In general, $\bar{X} * \bar{Y} \neq \bar{Y} * \bar{X}$, $\bar{X} \boxplus \bar{Y} \neq \bar{Y} \boxplus \bar{X}$, and $\bar{X} \boxminus \bar{Y} \neq \bar{Y} \boxminus \bar{X}$.

The operators O , E , L , and U (Odd, Even, Lower, and Upper) are defined by

$$(E\bar{X})_k = x_{2k}, \quad (L\bar{X})_k = x_k, \quad (5)$$

$$(O\bar{X})_k = x_{2k+1}, \quad (U\bar{X})_k = x_{N/2+k}$$

($k=0,1,\dots,N/2-1$)

These are illustrated below for the case $N = 8$.

$$E\bar{X} = \begin{bmatrix} x_0 \\ x_2 \\ x_4 \\ x_6 \end{bmatrix}, \quad O\bar{X} = \begin{bmatrix} x_1 \\ x_3 \\ x_5 \\ x_7 \end{bmatrix},$$

$$L\bar{X} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad U\bar{X} = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}.$$

Cyclic Convolution

Method of Bisection

The input sequences \bar{X} and \bar{Y} and the output sequence \bar{F} in the cyclic convolution $\bar{F} = \bar{X} \boxplus \bar{Y}$ can be partitioned into upper and lower parts.

$$\begin{aligned} L\bar{F} &= L\bar{X} * L\bar{Y} + U\bar{X} * U\bar{Y} + L\bar{X} \boxplus U\bar{Y} \\ &\quad + U\bar{X} \boxminus L\bar{Y} \\ U\bar{F} &= L\bar{X} * U\bar{Y} + U\bar{X} * L\bar{Y} + L\bar{X} \boxminus L\bar{Y} \\ &\quad + U\bar{X} \boxplus U\bar{Y} \end{aligned} \quad (6)$$

The origin of each term of (6) is clearly associated with one of the triangular areas shown in the example of Figure 3. Each term may be formally verified from the definitions (1)-(5). In the subsequent derivations we shall again use only the operator notation and visual term identification provided by a partitioned diagram such as Figure 3. In every case the validity of the operator equation can be verified for all positive n from the formal definitions.

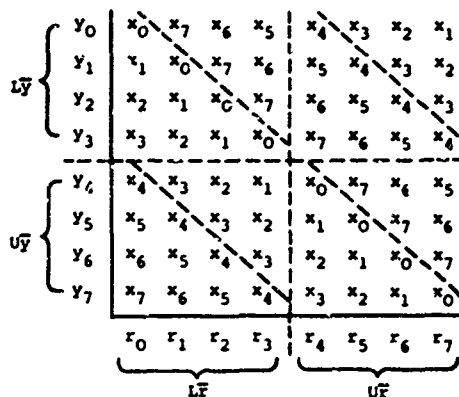


Figure 3. Decomposition of cyclic convolution $\tilde{F} = \tilde{X} \circledast \tilde{Y}$ by bisection for the case $N = 8$.

If the operator identity (4) is used to eliminate the complementary convolutions \circledast in (6),

$$\begin{aligned} L\tilde{F} &= (L\tilde{X} - U\tilde{X}) * (L\tilde{Y} - U\tilde{Y}) \\ &\quad + L\tilde{X} \circledast U\tilde{Y} + U\tilde{X} \circledast L\tilde{Y} \\ U\tilde{F} &= -(L\tilde{X} - U\tilde{X}) * (L\tilde{Y} - U\tilde{Y}) \\ &\quad + L\tilde{X} \circledast L\tilde{Y} + U\tilde{X} \circledast U\tilde{Y} \end{aligned} \quad (7)$$

Relations (7) can be expressed in terms of three sub-convolutions of sequences of length $N/2$. Let

$$\begin{aligned} \tilde{a} &= (L\tilde{X} + U\tilde{X}) \circledast (L\tilde{Y} + U\tilde{Y}), \\ \tilde{b} &= (L\tilde{X} - U\tilde{X}) \circledast (L\tilde{Y} - U\tilde{Y}), \\ \tilde{c} &= (L\tilde{X} - U\tilde{X}) * (L\tilde{Y} - U\tilde{Y}), \end{aligned} \quad (8)$$

then

$$\begin{aligned} L\tilde{F} &= \tilde{c} + 1/2 (\tilde{a} - \tilde{b}), \\ U\tilde{F} &= -\tilde{c} + 1/2 (\tilde{a} + \tilde{b}). \end{aligned} \quad (9)$$

Alternatively, (4) can be used to eliminate the linear convolutions $*$ in (6) and sub-convolutions defined,

$$\begin{aligned} \tilde{d} &= (L\tilde{X} + U\tilde{X}) \circledast (L\tilde{Y} + U\tilde{Y}), \\ \tilde{e} &= (L\tilde{X} - U\tilde{X}) \circledast (L\tilde{Y} - U\tilde{Y}), \\ \tilde{f} &= (L\tilde{X} - U\tilde{X}) \circledast (L\tilde{Y} - U\tilde{Y}), \end{aligned} \quad (10)$$

so that

$$\begin{aligned} L\tilde{F} &= 1/2 (\tilde{d} + \tilde{e}) - \tilde{f}, \\ U\tilde{F} &= 1/2 (\tilde{d} - \tilde{e}) + \tilde{f}. \end{aligned} \quad (11)$$

Results (9) and (10) illustrate that for each convolution type, cyclic or linear, and each derivational method, bisection or parisection, there is generally more

than one way to decompose the original convolution into three sub-convolutions. Each decomposition leads to a distinct algorithm.

Figure 4 illustrates one iteration in the A, B, and C-planes according to

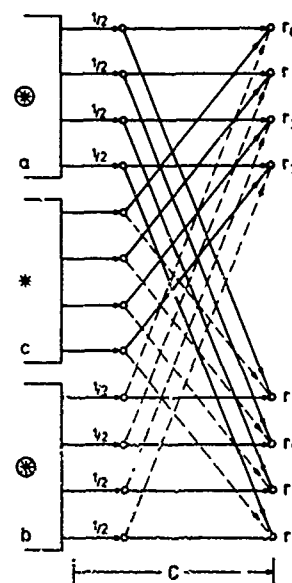
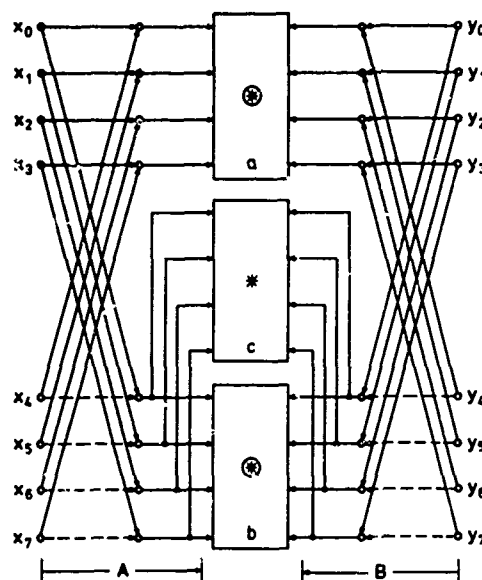


Figure 4. One iteration of the algorithm for cyclic convolution by bisection according to (8) and (9) for $N = 8$. Broken lines indicate negation. The scale factor $1/2$, shown adjacent to the appropriate lines, can be accomplished by shifting.

(8) and (9) for the case $N = 8$. To iterate further we must employ the linear convolution algorithm derived below. Figure 5 illustrates the complete A, B, and C-planes of the algorithm for $N = 8$. Bit-reversal of the order of sequences must be introduced in the complete algorithm at the transitions between cyclic sub-convolutions by bisection and linear sub-convolutions by parisection. There is no algorithm within the class for linear convolution by bisection. Hence the algorithm shown in Figure 5 is an hybrid of the bi- and parisection methods.

Method of Parisection

The input and output sequences of the cyclic convolution $\bar{Y} = \bar{X} \otimes \bar{Y}$ may likewise be partitioned into equal groups according to the index parity of the elements of the sequences.

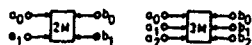
$$\begin{aligned} E\bar{Y} &= E\bar{X} \otimes E\bar{Y} + O\bar{X} \otimes O\bar{Y} \\ O\bar{Y} &= O\bar{X} \otimes (E\bar{Y})' + E\bar{X} \otimes O\bar{Y} \end{aligned} \quad (12)$$

This is illustrated in Figure 6. The prime denotes a cyclic shift of the elements of the sequence. If \bar{V} is of length M and $\bar{Z} = (\bar{V})'$, then

$$\begin{aligned} z_k &= v_{(k+1) \bmod M} \\ (k=0, 1, \dots, M-1) \end{aligned} \quad (13)$$

For the case $N = 8$,

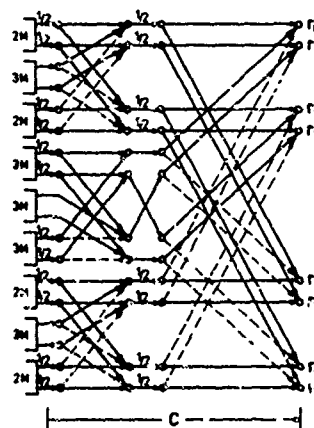
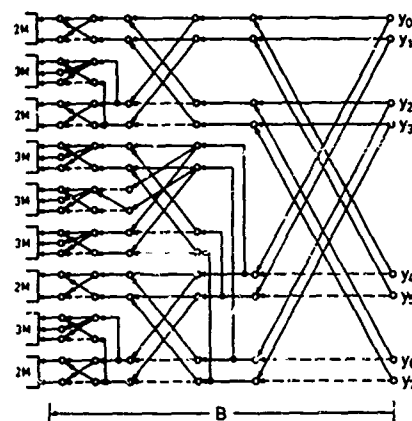
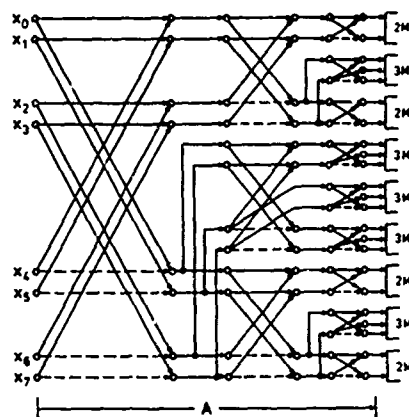
$$(E\bar{Y})' = \begin{bmatrix} y_2 \\ y_4 \\ y_6 \\ y_0 \end{bmatrix}$$



Boxes labelled 2M and 3M have two outputs each to the C-plane and perform two and three multiplications, respectively.

	2M	3M
OUTPUT-0	$\frac{1}{2}(a_0 b_0 + a_1 b_1)$	$\frac{1}{2}(a_0 b_0 + a_1 b_1)$
OUTPUT-1	$\frac{1}{2}(a_0 b_0 - a_1 b_1)$	$\frac{1}{2}(a_0 b_0 - a_1 b_1) - a_2 b_2$

Figure 5. Cyclic convolution by bisection for the case $N = 8$. The basic iteration is from (8) and (9), but the algorithm for linear convolution by parisection must be introduced to complete the flow.



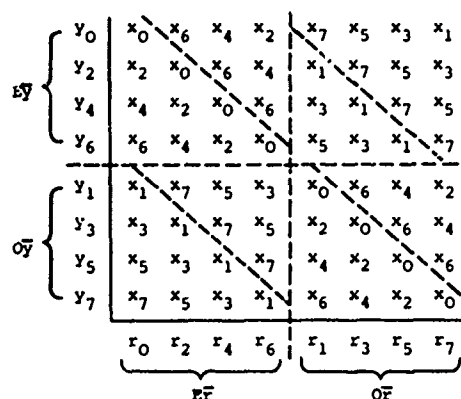


Figure 6. Decomposition of cyclic convolution $F = \bar{X} \otimes Y$ by parisection for $N = 8$.

If we choose the sub-convolutions

$$\begin{aligned} \bar{a} &= (E\bar{X} + O\bar{X}) \otimes (EY + OY), \\ \bar{b} &= (E\bar{X} - O\bar{X}) \otimes (EY - OY), \\ \bar{c} &= O\bar{X} \otimes [(EY)' - EY], \end{aligned} \quad (14)$$

then

$$\begin{aligned} EY &= 1/2 (\bar{a} + \bar{b}), \\ OY &= 1/2 (\bar{a} - \bar{b}) + \bar{c}. \end{aligned} \quad (15)$$

Figure 7 illustrates one iteration according to (14) and (15), and Figure 8 the complete algorithm, for $N = 8$. Parisection results in bit-reversed input and output sequences in the complete algorithm. We shall call the operations which can be identified with $(EY)' - EY$ in \bar{c} at each level of iteration a "slipping" algorithm. It is seen clearly at the input to \bar{c} on plane B of Figure 7. Bit-reversed sequences in the complete algorithm necessitate performing the slipping algorithms at all levels of iteration in bit-reversed order. Though the flow of the bit-reversed slipping algorithm appears unacceptably complex, it in fact can be performed simply and in-place as described in Appendix A.

The corrections \bar{c} are not required for cyclic convolution of sequences of length 2 and hence they do not appear in the inputs to the multipliers $2M$ in Figure 8. Matters are further simplified at the final level of iteration because the slipping algorithm $(\bar{V})' - \bar{V}$ is degenerate for length-2 sequences. Consequently the mid-multiplier of each group of three in Figure 8 is of the type $1M$.

An alternate algorithm for cyclic convolution by parisection is derived using the identity

$$\bar{W} \otimes (\bar{V})' = (\bar{W} \otimes \bar{V})'$$

in (14) to rewrite \bar{c} .

$$\bar{c} = (O\bar{X} \otimes EY)' - O\bar{X} \otimes EY. \quad (16)$$

This is simply the slipping algorithm applied to $O\bar{X} \otimes EY$. Figures 9 and 10 illustrate one iteration and the com-

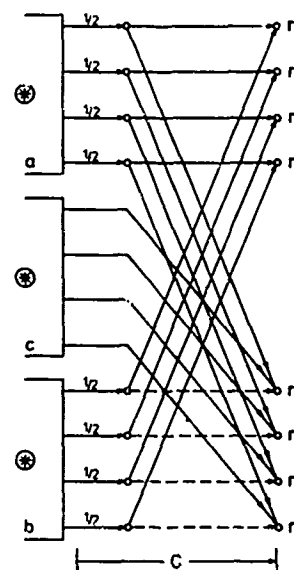
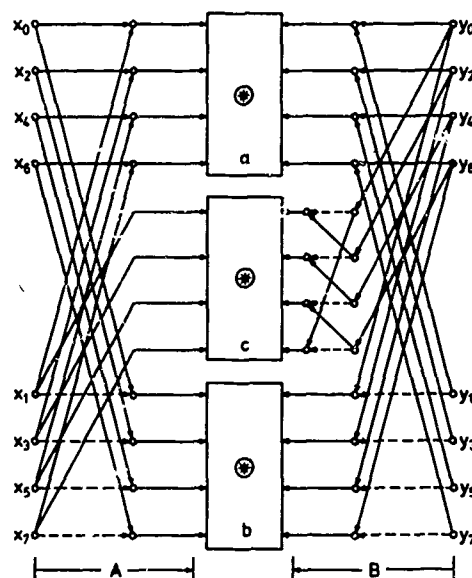
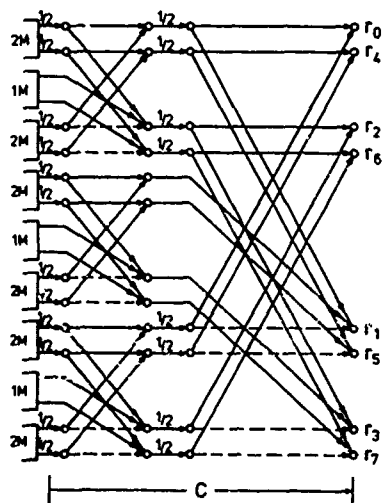
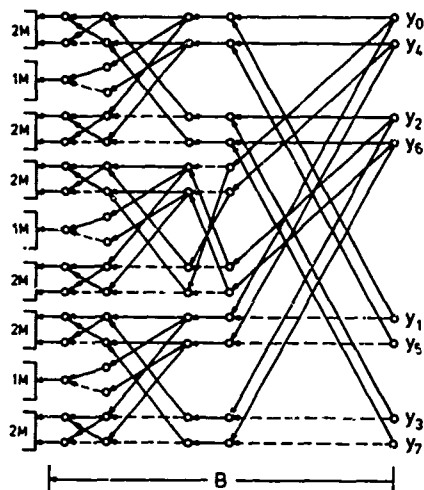
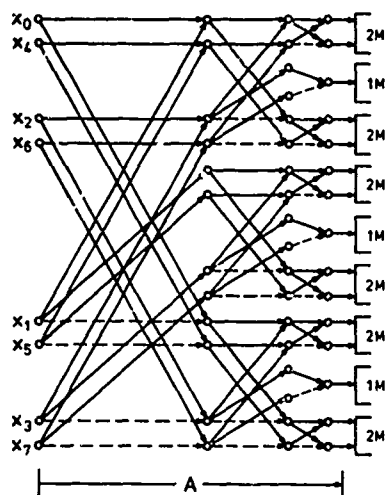


Figure 7. One iteration of the algorithm for cyclic convolution by parisection according to (14) and (15) for $N = 8$.



plete algorithm for $N = 8$ based on \bar{a} and \bar{b} from (14) and \bar{c} from (16).

This algorithm is in fact that chosen by Pitassi for illustration. As he showed, it has the advantage that the outputs of the A and B-planes can be simply related to the Walsh transforms of \bar{x} and \bar{y} with commensurate savings of storage.

Linear Convolution

Method of Parisection

Figure 11 illustrates the linear convolution $\bar{F} = \bar{x} * \bar{y}$ by parisection for $N = 8$. The corresponding relations are

$$\begin{aligned} E\bar{F} &= E\bar{x} * E\bar{y} + O\bar{x} * O\bar{y}, \\ O\bar{F} &= O\bar{x} * (E\bar{y})'' + E\bar{x} * O\bar{y}. \end{aligned} \quad (17)$$

Double prime denotes a shift of the elements of a sequence, with zero filling the vacated position. If \bar{v} is of length M and $\bar{z} = (\bar{v})''$, then

$$z_k = v_{k+1}, \quad (k=0, 1, \dots, M-2)$$

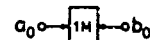
$$z_{M-1} = 0.$$

For the case $N = 8$,

$$(E\bar{y})'' = \begin{bmatrix} y_2 \\ y_4 \\ y_6 \\ 0 \end{bmatrix}$$

We may choose sub-convolutions analogous to (14) by replacing \oplus by $*$ and \ominus by $-$.

$$\begin{aligned} \bar{a} &= (E\bar{x} + O\bar{x}) * (E\bar{y} + O\bar{y}), \\ \bar{b} &= (E\bar{x} - O\bar{x}) * (E\bar{y} - O\bar{y}), \\ \bar{c} &= O\bar{x} * [(E\bar{y})'' - E\bar{y}]. \end{aligned} \quad (18)$$



Boxes labelled 1M have two outputs to the C-plane and perform one multiplication.

OUTPUT-0	$-c_0 b_0$
OUTPUT-1	$+a_0 b_0$

Figure 8. Cyclic convolution by parisection. Complete A, B, and C-plane flow using (14) and (15) for the case $N = 8$.

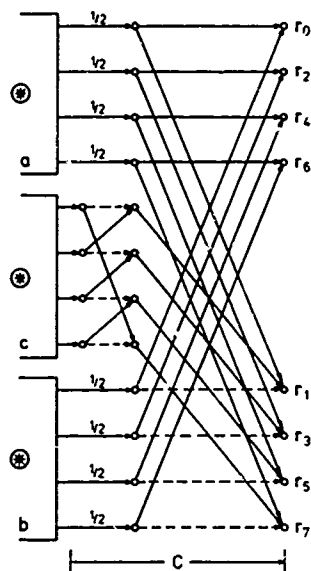
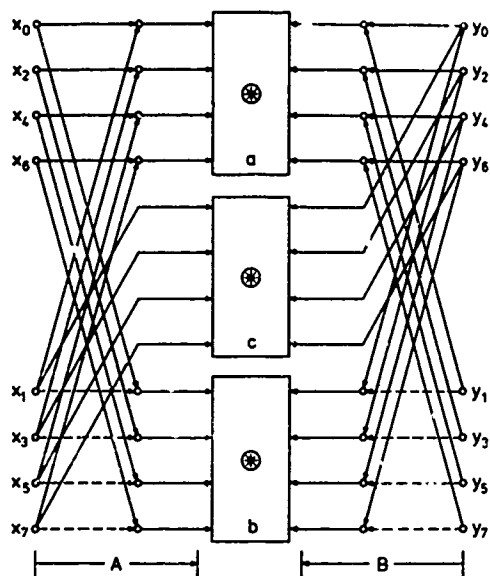
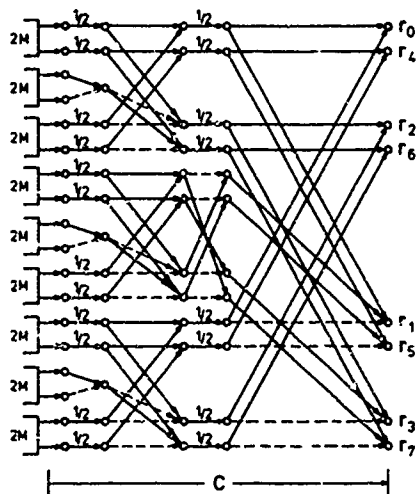
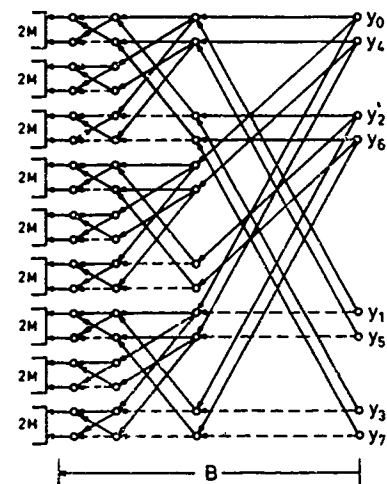
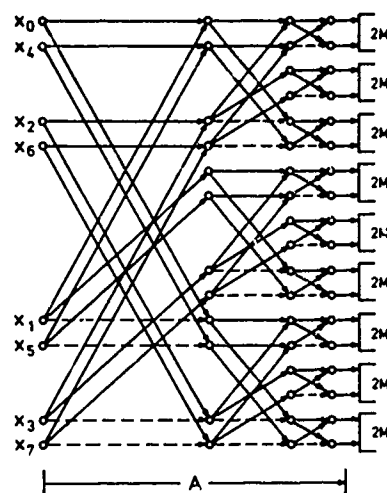


Figure 9. One iteration of the algorithm for cyclic convolution by parisection according to \bar{a} and \bar{b} from (14) and \bar{c} from (16) for the case $N = 8$.

Figure 10. Cyclic convolution by parisection. Complete A, B, and C-plane flow using \bar{a} and \bar{b} from (14) and \bar{c} from (16) for $N = 8$. Note the simplification of the slipping algorithm at the output of multipliers 2, 5, and 8 in the C-plane.



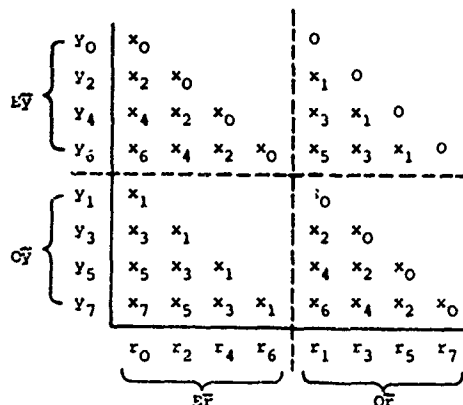


Figure 11. Decomposition of linear convolution $\bar{F} = \bar{X} * \bar{Y}$ by parisection for $N = 8$.

Or, we may use the identity

$$\bar{W} * (\bar{V})'' = (\bar{W} * \bar{V})''$$

to rewrite \bar{C} .

$$\bar{C} = (\bar{O}\bar{X} * \bar{E}\bar{Y})'' - \bar{O}\bar{X} * \bar{E}\bar{Y}. \quad (19)$$

In either case, (15) applies.

The form $(\bar{V})'' - \bar{V}$ is a slipping algorithm with the main (logically forward) diagonal removed. Hence, one-iteration diagrams analogous to Figures 7 and 9 apply, but with \oplus replaced by $*$ and the slipping algorithm main diagonal removed. Figure 12 illustrates the complete algorithm based on (15) and (18) for $N = 8$. Note that three multiplications are required per linear sub-convolution of length two and that the modified slipping algorithm in the final stage of iteration does not simplify as for cyclic convolution.

Method of Bisection

Figure 13 illustrates the partitioning of the linear convolution $\bar{F} = \bar{X} * \bar{Y}$ by bisection. The relevant equations are

$$\begin{aligned} L\bar{F} &= L\bar{X} * L\bar{Y} + L\bar{X} \boxplus U\bar{Y} + U\bar{X} * U\bar{Y}, \\ U\bar{F} &= L\bar{X} * U\bar{Y}. \end{aligned} \quad (20)$$

Evidently these equations can not be reduced to simple combinations of three sub-convolutions.

Number of Multiplications

Each level of iteration increases the total number of sub-convolutions by a

power of 3 in all of the algorithms. There can be as many iterations as the number of times the input sequences can be halved. If $N = 2^n$, 3^{n-1} sub-convolutions of length 2 are required in the completed algorithm. One, two, or three [generally complex] multiplications are required by each such sub-convolution. The multiplications will be real if the sequences to be convolved are real.

The final sub-convolutions for linear convolution by parisection all require 3 multiplications (Figure 12), so that the total number of multiplications is

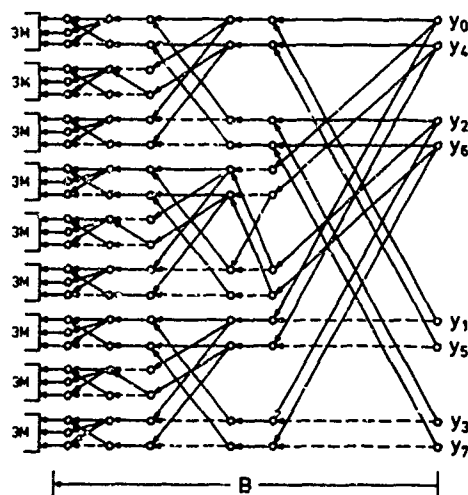
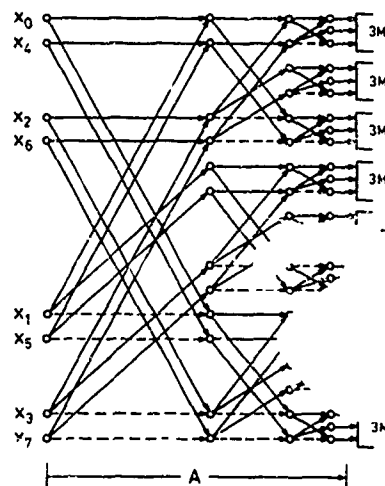


Figure 12. Linear convolution by parisection. Complete A and B-plane flow using (15) and (18) for the case $N = 8$. Plane C is as shown in Figure 8 for cyclic convolution, but with all multipliers of the 3M type.

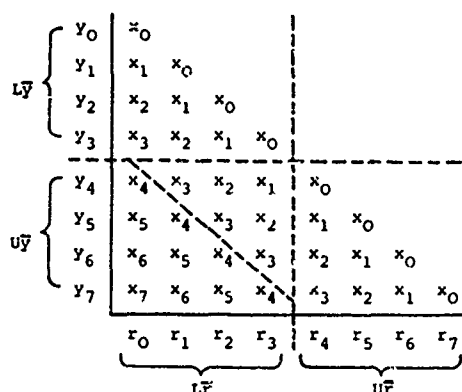


Figure 13. Decomposition of linear convolution $F = \bar{X} * \bar{Y}$ by bisection for the case $N = 8$.

3^n . Cyclic convolution by parisection can be performed in two ways. If the slipping algorithms are in the C-plane, all final sub-convolutions require 2 multiplications (Figure 10). The total number is therefore $2 \cdot 3^{n-1}$. If the slipping algorithms are in the B-plane (Figure 8), the mid-convolution in each group of three will require only one multiplication, so the total number is reduced to $5 \cdot 3^{n-2}$.

At each level of the hybrid algorithm for cyclic convolution by bisection (Figure 5), the number of sub-convolutions of the cyclic type increases by a power of 2. In the complete algorithm there are 2^{n-1} cyclic sub-convolutions of length 2, each requiring 2 multiplications. The remaining $3^{n-1} - 2^{n-1}$ sub-convolutions are of the linear type requiring 3 multiplications. The total number of multiplications is therefore $3^n - 2^{n-1}$. These results are summarized in Table 1 below.

Linear convolution can be effected via a double-length cyclic convolution padded with zeros. Table 1 reveals that

Conv.	Mult.	Method	Relevant Eq.
cyc	$5 \cdot 3^{n-2}$	parisect	(14), (15)
cyc	$2 \cdot 3^{n-1}$	parisect	(14), (15), (16)
cyc	$3^n - 2^{n-1}$	bisect	(8), (9)
lin	3^n	parisect	(15), (18)

Table 1. Summary of convolution algorithms.

linear convolution using (15) and (18) is superior.

Partial Convolution

The algorithms by parisection are advantageous when only a segment of the complete convolution is required, if the slipping algorithms are not in the C-plane. Let us assume that the first $M = 2^m$ points of the convolution are to be evaluated ($m < n$). Inspection of Figures 7 and 8 reveals that sub-convolutions yielding the first 2^{m-1} points are required at the first level of iteration. At the k -th level of iteration, 3^k sub-convolutions of sequences of length 2^{n-k} are required yielding 2^{m-k} points. When $k = m$, each sub-convolution must yield only its first point. If each point required at the k -th iteration level is evaluated in the straightforward way, $N(3/2)^m$ multiplications are involved. This number is less than that required for full convolution by parisection, except for cyclic convolution using (14) and (15) with no exception if $m < n - 1$.

Thus partial cyclic linear convolutions based on equations (14), (15), and (18), using algorithms tailored as described above, all require $N(3/2)^m$ multiplications. Partial cyclic and linear convolutions using the FFT technique can be evaluated in approximately $2N(n+m+4) - 4M$ and $4N(n+m+5) - 4M$ real multiplications respectively ($m < n$). These estimates can be improved slightly with careful programming. It is assumed that each stage of the inverse transform leading to the partial convolution is evaluated only to the extent required for partial results. Linear convolution by the FFT method is assumed to be accomplished by a full doubling of the input sequences by zero-padding, though less would suffice for partial results, because one can not otherwise easily predict the prime factors of the new sequence length.

The algorithms described here for partial cyclic and linear convolution require fewer multiplications than the FFT technique for suitable combinations of m and n . Table 2 lists, using the above estimates, some partial convolution lengths M and the corresponding input sequence lengths N for which partial convolution can be evaluated more efficiently, in terms of real multiplications, than with the FFT. It is assumed that the input sequences are real so that the new algorithms require $N(3/2)^m$ real multiplications.

M	N
2^k	$\geq 2^{k+1}$, ($k \leq 9$, cyclic conv.) ($k \leq 11$, linear conv.)
2^{10}	$\geq 2^{15}$, cyclic conv.
2^{11}	$\geq 2^{29}$, cyclic conv.
2^{12}	$\geq 2^{49}$, cyclic conv. $\geq 2^{16}$, linear conv.

Table 2. Some trade-off values for partial convolution using the new algorithms. See text for explanation.

The new algorithms are particularly advantageous in applications which require convolution (or correlation) of very long sequences, but in which only the first relatively few terms of the convolution are required. One such application is the estimation of the power-density spectrum of noise from its autocorrelation.

Relation to Walsh Transform

Inspection of the complete A and B-plane diagrams for all algorithms in the class reveals four basic flow patterns. The slipping algorithms, which appear in the derivations by parisection, can always be transferred from the B to the C-plane as previously shown and so are not considered here.

The rectangles in Figure 14 signify that the associated nodal variables are, in general, column vectors of length greater than one. S is Shanks' algorithm [4] of the appropriate order for computing the Walsh transform. The arrows \Rightarrow indicate the transformations which are applied to the nodal variables to yield

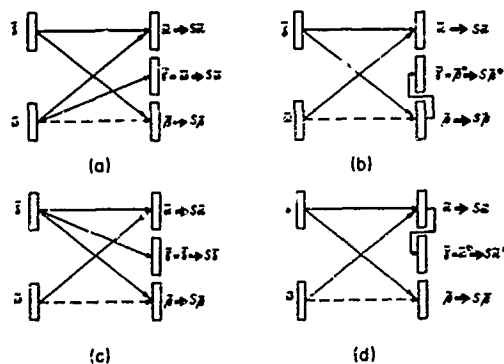


Figure 14. Basic A-plane (a,b) and B-plane (c,d) flow patterns occurring in the derivations by parisection (a,c) and by bisection (b,d).

the corresponding nodes in the output of the A and B-planes. Super * indicates that the corresponding column vector is to be placed in bit-reversed order.

If the flow associated with $\bar{\gamma}$ at all levels of iteration is eliminated, the A and B-planes reduce to the flow pattern of Shanks' algorithm. The input sequence is in natural order for convolution algorithms proceeding by bisection, and in bit-reversed order for those by parisection. The latter is equivalent to the canonical form of Shanks' algorithm discussed in Appendix B.

Shanks' algorithm yields, in bit-reversed order, the Walsh coefficients of a naturally ordered input sequence.

$$S\bar{S} = (W\bar{S})^*$$

where W is the Walsh transform operator. In Appendix B we show that Shanks' algorithm applied to a bit-reversed sequence yields the Walsh coefficients in natural order.

$$S\bar{S}^* = W\bar{S} = (S\bar{S})^*$$

The nodes in the output associated horizontally with $\bar{\gamma}$ at any level of iteration can be derived from the much smaller set of nodal values produced by Shanks' algorithm applied to the input sequence. Hence the convolution algorithms can be implemented with great savings of storage. For parisection (see Figure 14)

$$\bar{\alpha} = \bar{\delta} + \bar{\omega}, \quad \bar{\beta} = \bar{\delta} - \bar{\omega}$$

from which

$$\begin{aligned} \bar{\omega} &= 1/2(\bar{\alpha} - \bar{\beta}), & S\bar{\omega} &= 1/2(S\bar{\alpha} - S\bar{\beta}), \\ \bar{\delta} &= 1/2(\bar{\alpha} + \bar{\beta}), & S\bar{\delta} &= 1/2(S\bar{\alpha} + S\bar{\beta}). \end{aligned} \quad (21)$$

For bisection we need only

$$S\bar{S}^* = (S\bar{S})^*, \quad S\bar{\alpha}^* = (S\bar{\alpha})^* \quad (22)$$

$S\bar{\omega}$ and $S\bar{\delta}$ are available as sub-sequences of the N nodal values produced by Shanks' algorithm. $S\bar{\omega}$, $S\bar{\delta}$, $S\bar{S}^*$, and $S\bar{\alpha}^*$ can be computed from $S\bar{\alpha}$ and $S\bar{S}$ as needed using (21) and (22).

Appendix A

Bit-reversed Slipping Algorithm

The slipping algorithm can be performed in bit-reversed order in-place and without an extra storage array. For the case $N = 8$ the steps are

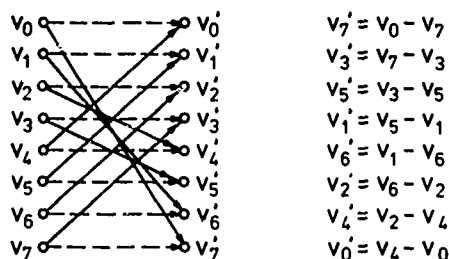


Figure A1. Slipping algorithm in bit-reversed order for the case $N = 8$.

The first step is simply $v_7' = -v_7$ if the slipping is applied to a linear convolution. If the steps are performed in the order indicated, only one word is required to temporarily store the previous contents of the register just overwritten. The required indices are just the bit-reversed integers in reverse order. They can be generated as follows.

Integers are represented by an n -bit word. Begin with an n -bit word of 1's ($N - 1$). This is the first required index. The next index is found by:

- a) 1's complementing the left-most 1 in the n -bit word.
- b) 1's complementing any zeros which may have existed to the left of the bit position complemented in a).

Subsequent indices are generated by applying steps a) and b) to the most recent index. The process is repeated until zero results. This procedure is particularly convenient in machines with normalizing and arithmetic shift instructions.

Appendix B

Shanks' algorithm applied to a bit-reversed sequence yields the Walsh coefficients of the sequence in natural order. To show this we first show that the Walsh transform of a bit-reversed sequence is the Walsh transform in bit-reversed order. Let the Walsh transform of \bar{a} be $\bar{\beta}$.

$$\bar{\beta} = W\bar{a} \quad (B1)$$

The elements of $\bar{\beta}$ are

$$\beta_k = \sum_{m=0}^{N-1} \alpha_m W_{mk} = \sum_{m=0}^{N-1} \alpha_m \prod_{i=0}^{n-1} (-1)^{m_{n-1-i} k_i} \quad (B2)$$

($k=0, 1, \dots, N-1$)

W_{mk} is the mk -th element of the matrix representing the Walsh transform. m_i and k_i are the i -th digits of the n -bit binary representations of m and k .

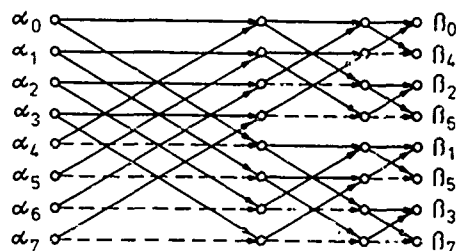
$$m = 2^{n-1} m_{n-1} + 2^{n-2} m_{n-2} + \dots + 2m_1 + m_0 \quad (B3)$$

$$k = 2^{n-1} k_{n-1} + 2^{n-2} k_{n-2} + \dots + 2k_1 + k_0$$

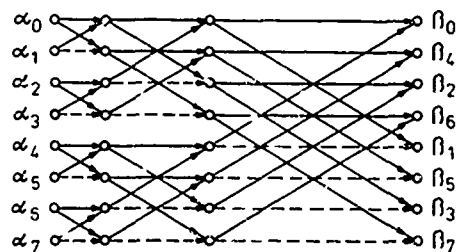
With super * representing bit-reversed order, it is clear that

$$\beta_k = \sum_{m=0}^{N-1} \alpha_m^* \prod_{i=0}^{n-1} (-1)^{m_i k_i} \quad (B4)$$

and also that



(a)



(b)

Figure B1. a) Shanks' algorithm S for Walsh coefficients when $N = 8$.

b) Canonical form of Shanks' algorithm S' for $N = 8$.

$$\beta_k^* = \sum_{m=0}^{N-1} \alpha_m^* \prod_{i=0}^{n-1} (-1)^{m_i k_{n-1-i}} \quad (B5)$$

Introduce the change of variable $p = n-1-i$,

$$\beta_k^* = \sum_{m=0}^{N-1} \alpha_m^* \prod_{p=0}^{n-1} (-1)^{m_{n-1-p} k_p} \quad (B6)$$

Hence

$$\bar{\beta}^* = W\bar{\alpha}^* = (W\bar{\alpha})^* \quad (B7)$$

Shanks' algorithm applied to a naturally ordered sequence yields the Walsh coefficients in bit-reversed order [4].

$$S\bar{\alpha} = (W\bar{\alpha})^* = W\bar{\alpha}^*$$

If we replace $\bar{\alpha}$ by $\bar{\alpha}^*$,

$$S\bar{\alpha}^* = W\bar{\alpha} = (S\bar{\alpha})^* \quad (B8)$$

as was to be shown.

Relation (B8) implies a canonical form of Shanks' algorithm (see Figure B1). Let S be applied to the sequence $\bar{\alpha}^*$ and let the input sequence then be put into natural order without detaching any arrows in the flow diagram. Horizontally

associated nodes are to remain so. There will result

$$S'\bar{\alpha} = W\bar{\alpha}^* = S\bar{\alpha} \quad (B9)$$

S' is the canonical form of Shanks' algorithm which bears a relation to S similar to that between the canonical forms of the FFT [3].

References

1. D. A. Pitassi, "Fast Convolution Using the Walsh Transform", Applications of Walsh Functions, 1971 Proceedings, pp. 130-133, April 1971.
2. In this paper bisection denotes separation of a sequence into two groups according to the value of the position index relative to the midpoint of the sequence. I.e., into groups of high and low index. Parisection denotes separation into two groups according to the parity of the position index. I.e., into groups of odd and even index.
3. W. T. Cochran, J. W. Cooley, D. L. Favon, H. D. Helms, R. A. Kaenel, W. Lang, G. C. Maling, Jr., D. E. Nelson, C. M. Rader, and P. D. Welch, "What is the Fast Fourier Transform?", Proc. IEEE, vol. 55, pp. 1664-1674, October 1967.
4. J. L. Shanks, "Computation of the Fast Walsh-Fourier Transform", IEEE Trans. Computers, vol. C-18, pp. 457-459, May 1969.

LINEAR MINIMUM MEAN-SQUARE ERROR CODES

G. R. Redinbo
Department of Electrical Engineering
University of Wisconsin
Madison, Wisconsin

Abstract

We consider the use of (n,k) binary group codes over a channel from the viewpoint of a mean-square error criterion. It is assumed that the transition probabilities of the channel satisfy an additive property. The elements of a generic code are labeled by the first 2^k nonnegative integers, $0, 1, \dots, 2^k - 1$, so as to obtain integer-valued random variables at the input and output. Optimum performance is judged by the mean-square of the overall error between input and output. Based upon this criterion the optimum linear encoding and decoding rules as well as the optimum group code are determined simultaneously.

Our analysis and synthesis uses the Walsh transforms of the channel transition probabilities. It is shown that the optimum rules and code are completely specified by k linearly independent Walsh functions defined on the binary n -dimensional vector space. Unfortunately the most significant part of this result is that the optimum rules are always projection mappings, i.e., rules which use none of the parity-check positions.

Introduction

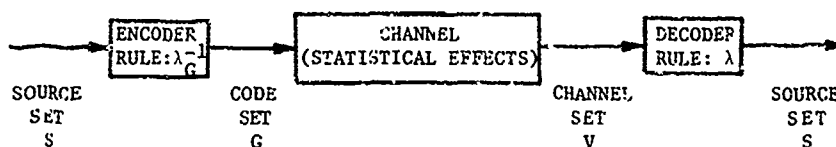
This paper presents the development of classes of algebraic codes using the Walsh analysis of the statistics of the system in which they are to be employed. This approach adds to the algebraic and the dimension of Fourier analysis. As in the recent articles by Crimmins, et al., [1] and Crimmins and Horwitz [2] we will use the mean-square of the overall error as a criterion of goodness. However unlike either of these papers the minimization of this criterion will also dictate the linear code as well as the linear encoding and decoding rules to implement it. Our results have a negative aspect to them in that the optimum rules are just projection mappings. This is undesirable for coding purposes since projection mappings are equivalent to rules which do not use the parity-check positions in a code.

A model of the system which we will consider is shown in the figure. The source alphabet will be taken as the set of the first 2^k nonnegative integers, i.e., $S = \{0, 1, \dots, 2^k - 1\}$. The purpose of this choice of a source set is to give real-valued random variables for use in the mean-square error criterion. The encoder mechanizes the mapping rule λ_G^{-1} (the meaning of the subscript and superscript will be clear shortly) and transforms the message (an integer from S) into a member of an n -dimensional binary vector space V . Since we are seeking a linear code, every member of S is carried onto a member of some k -dimensional subspace G of V . At this point we may only hypothesize the existence of such a code G ; its exact specification is to be determined below. Clearly this mapping λ_G^{-1} must be in a one-to-one fashion for otherwise the decoder which reverses this operation would be confronted with ambiguities. The inverse of the encoding rule is a rule λ_G which is one-to-one from the subspace G onto S .

The statistical effects of the channel can transform a symbol from G into a member of V which is not in the subspace G . Therefore the decoder involves a rule λ which maps from V back into the source set S . Since the decoder must reverse the action of the encoder, the mapping λ_G^{-1} must be the inverse of the decoder rule λ when it is restricted to act only on the subspace G . The purpose of the subscript and superscript in λ_G^{-1} is now clear.

Thus the problem is to find the mapping λ such that a minimum mean-square error is achieved. In turn, the mapping λ defines the subspace G , the linear block code. Our approach will use the Fourier transforms of the channel transition probabilities. However the character group of the binary vector space V , i.e., the kernels for the transforms, are exactly the first 2^n Walsh functions suitably defined on V . Therefore the abstract Fourier transforms become the Walsh transforms.

Preliminaries



COMMUNICATION SYSTEM MODEL

any of the well known properties of the Walsh Transforms will be used throughout this paper. Also since the subject of Abstract Fourier Analysis is well covered in several books [3-6], we will use standard notation and terminology without specific reference.

We will introduce several additional symbols. The set S which was defined as, $S = \{0, 1, \dots, 2^k - 1\}$ will also be endowed with an additive operation \oplus which is dyadic addition (component-wise modulo 2 addition - no carries). V is an n -dimensional binary vector space with character group V^* . G is a k -dimensional subspace of V with annihilator A^* , i.e., $A^* = \{\chi \in V^* : \chi(g) = 1 \text{ for all } g \in G\}$. If $f(v)$ is a complex valued function on V , its transform is:

$$\hat{f}(\chi) = \sum_{v \in V} f(v) \chi(v) ; \chi \in V^* \quad (1)$$

The sum is over all 2^n elements of V . The inverse transform is:

$$f(v) = \frac{1}{2^n} \sum_{\chi \in V^*} \hat{f}(\chi) \chi(v) ; v \in V \quad (2)$$

This sum is over all 2^n elements of V^* .

We will assume that the probability distribution of the source is uniform, i.e., $p(s) = \frac{1}{2^k}$ for all $s \in S$. We will further assume that the channel transition probabilities are given and that:

$$P(v|g) = P(v-g|0) \quad g \in G, v \in V, 0 = \text{identity} \quad (3)$$

The following lemma gives a characterization of the class of mapping rules of the desired type. In this lemma we also distinguish the subclass of projection mappings, i.e., linear mappings which always take some basis elements of V into zero. A projection mapping is equivalent to a rule which does not use all available parity-check positions in a code.

Lemma

There is a linear mapping $\lambda: V \rightarrow S$ whose restriction to a subgroup G , $\lambda|_G$, is an isomorphism if and only if there is a basis for V^*/A^* , $\{\chi_j\}_{j=1}^k$ and a basis for A^* , $\{\chi_j\}_{j=k+1}^n$ and binary constants $a_{ij} = 0, 1$, where $1 \leq i \leq k$, and $k+1 \leq j \leq n$ such that

$$\lambda = \sum_{i=1}^k 2^{k-i} \left[\frac{1 - \chi_i \prod_{j=k+1}^n \chi_j^{a_{ij}}}{2} \right] \quad (4)$$

For a given G , this representation is unique. Furthermore λ is also a projection mapping taking m basis elements of V into zero if and only if there are m indices, j_1, j_2, \dots, j_m , for which $a_{ij_j} = 0$ for all $i = 1, 2, \dots, k$.

The proof of this lemma is offered in the appendix.

The Result

The main result is contained in the following theorem

Theorem: The necessary and sufficient conditions for a linear mapping $\lambda: V \rightarrow S$ which is an isomorphism on a k -dimensional subspace G to achieve a minimum mean-square error over all possible choices of such mappings and k -dimensional subspaces are given as:

There is a set of linearly independent characters $\{\chi_i\}_{i=1}^n$ satisfying the conditions,

$$i) \quad 1 \in \bigcap_{j=k+1}^n \text{Ker } \chi_j \quad (5)$$

$$ii) \quad \lambda = \sum_{i=1}^k 2^{k-i-1} (1 - \chi_i) \quad (6)$$

$$iii) \quad \hat{P}(\chi_i) \geq \hat{P}(\beta) \quad (7)$$

for all $\beta \in \{\chi_1, \chi_2, \dots, \chi_{i-1}\}$

$i = 1, 2, \dots, k$.

The minimum mean-square error is:

$$\epsilon^2 = \frac{4^k - 1}{6} - 2^{2k-1} \sum_{r=1}^k \frac{1}{4^r} \hat{P}(\chi_r) \quad (8)$$

The proof of this results is also contained in the appendix. Note there is no uniqueness claim associated with the optimum mapping.

Comments

The significant part of this result is that the optimum rules are just projection mappings. This may be seen by combining equation (6) with the last part of the lemma. Hence in effect only the assignment of elements of S to the code members and the choice of G as a subspace are effected by the MSE criterion. However if this is the desired result, the sufficient conditions of the theorem are easy to apply. One must compute a finite number of transforms, order them, and extract the characters according to inequality (7). These determine the rule λ and it is possible to invert it and thus find the encoding rule $\lambda|_G^{-1}$. We can also determine the code G and compute the operating mean-square error if we wish.

What allows us to bridge the sometimes large gap between theory and true implementation is the fact that V^* corresponds to suitably defined Walsh functions. Hence the abstract transforms are the well known Fine transforms [7] which employ Walsh functions in their kernels.

The exact nature of the transform matters little if we are designing a fixed system be-

¹ Ker χ_j is the kernel of the homomorphism χ_j .

² The symbol $\{\chi_1, \dots, \chi_m\}$ denotes the subspace of V^* spanned by these characters.

cause the processing time for the transforms is not critical. However one possible use of these codes is in an adaptive system. Since the Walsh transforms of the channel statistics determine the coding rules, the equipment required for the real-time processing in an adaptive receiver is not only feasible but practical. If we were using a one-way channel a processor at the receiver could alter the decoding rule as to sustain optimum or near optimum MMSE performance under the constraint that the encoding rule remain fixed. However if a noiseless feedback path were also available both rules and the code could be varied.

The Transform

We note the following property.

Property The character group V^* is isomorphic with the set of 2^n Walsh functions.

The proof is obvious once a suitable definition of the Walsh function on V is given. Suppose $u \in V$ is represented as $u = (u_0, u_1, \dots, u_{n-1})$ and let m , an integer between 0 and $2^n - 1$ have a dyadic expansion as: $m = m_0 2^0 + m_1 2^1 + \dots + m_{n-1} 2^{n-1}$ where $m_i = 0, 1$. A definition compatible with Fine's [7] is:

$$w_m(u) = (-1)^{\sum_{i=0}^{n-1} m_i u_i} \quad (9)$$

This transform may also be displayed using matrix and vector notation. It will be convenient to index each member of V by its equivalent radix 2 representation, i.e., v_j where $j = v_0 2^0 + v_1 2^1 + \dots + v_{n-1} 2^{n-1}$ corresponds to the n -tuple $(v_0, v_1, \dots, v_{n-1})$. We will define the two column vectors \underline{P} and $\hat{\underline{P}}$ as:

$$\underline{P}^T = (P(v_0|0), P(v_1|0), \dots, P(v_{2^n-1}|0)) \quad (10a)$$

$$\hat{\underline{P}}^T = (P(w_0), P(w_1), \dots, P(w_{2^n-1})) \quad (10b)$$

We form the matrix H by defining each of its rows m as the vector representation of $w_m(v)$.

Hence each element h_{mj} of H is $w_m(v_j)$ and it is easy to show that H is a 2^n th order Hadamard matrix. So the transforms can be obtained from

$$\hat{\underline{P}} = H \underline{P} \quad (11)$$

Examples

We will present two examples. In the first example we will consider a Binary Symmetric Channel (BSC) with bit error probability p . Let $q = 1 - p$, and assume that $q > p$. For this case \underline{P} is of the following form:

$$\underline{P}^T = (q^n, pq^{n-1}, pq^{n-1}, p^2 q^{n-2}, pq^{n-1}, p^2 q^{n-2}, pq^{n-1}, p^3 q^{n-3}, \dots, p^{n-1} q, p^n) \quad (12)$$

If we denote the elements of \underline{P} by p_i where

$i = 0, 1, \dots, 2^n - 1$, we make the observation that $p_{i+2^r} = p_i(pq^{-1})$ for $r = 0, 1, \dots, n-1$. So if we have \underline{P} constructed in the first 2^S position, the next 2^S can be obtained from the first by multiplying by (pq^{-1}) and linearly translating. Coupling this with the symmetry in H , the vector $\underline{H} \underline{P}$ can be obtained by summing the rows of a matrix Q which is defined by the following matrix product.

$$Q = \begin{bmatrix} q^n & pq^{n-1} \\ q^n & -pq^{n-1} \end{bmatrix} \otimes \begin{bmatrix} n-1 \\ \otimes \end{bmatrix} \begin{bmatrix} +1 & +pq \\ +1 & -pq^{-1} \end{bmatrix} \quad (13)$$

The symbol \otimes denotes the Kronecker matrix product whereas the symbol $\begin{bmatrix} n-1 \\ \otimes \end{bmatrix}$ means Kronecker matrix product $(n-1)$ times. Here of course $n > 2$. Therefore the vector of the transform, $\hat{\underline{P}}$ satisfies:

$$\hat{\underline{P}} = Q \underline{P} \quad (14)$$

$\underline{1}$ is the column vector with 2^n ones. It easily follows that

$$\hat{P}(w_m) = (q-p)^{n-z} \quad (15)$$

z is the number of zeros in the dyadic expansion of the index m .

Since we have an analytic expression for the transforms we will apply the theorem to the general case of an (n, k) binary code. The linearly independent Walsh functions as determined by part (iii) of the theorem are those with indices which are powers of two. Obviously the choices are not unique. The MMSE is easily computed to be

$$\epsilon^2 = \frac{p}{4} (4^k - 1) \quad (16)$$

Our second example concerns a compound channel, i.e., a channel with both bit and burst errors. At this time there is no known analytic expression for the transforms of its transition probabilities. The statistical description for this example is described as follows. The probability of no errors in a word of length n is ϵ_0 while the probability of a single bit being in error is ϵ_1 . We assume that the one error is equally likely in any position. Hence $P(0|0) = \epsilon_0$ and $P(v|0) = \frac{\epsilon_1}{n}$ where v has only one nonzero component in its n -tuple representation. If there is more than one error we assume that it is the result of a burst error contained wholly within the received word and that each burst is equally likely for all possible configurations. Thus the distribution of the burst length b is linear and may be described by the following expression.

$$P(b) = \frac{2(1-\epsilon_0-\epsilon_1)}{n(n-1)} (n+1-b) \quad 2 \leq b \leq n \quad (17)$$

Hence the channel transition probabilities for vectors v which have a single burst of length b in them are given by

$$P(v|0) = \frac{P(b)}{2^{b-2}(n-b+1)} \quad \begin{matrix} 2 \leq b \leq n \\ v \text{ has burst length } b \end{matrix} \quad (18)$$

For this channel model with $n=7$, $\epsilon_0=0.950$ and

$\epsilon_1=0.025$ we have computed the necessary transforms. For a (7,4) code one choice of the optimum code in terms of the theorem is given by: $x_1^{w_1}, x_2^{w_2}, x_3^{w_3},$ and $x_4^{w_4}$. The mean-square error is 0.926. However for a (7,3) code we can make the following choices: $x_1^{w_1}, x_2^{w_2},$ and $x_3^{w_3}$. In this case the MSE is 0.229.

References

1. T.R. Crimmins, H.M. Horwitz, C.J. Palermo, and R.V. Palermo, "Minimization of Mean-Square Error for Data Transmitted Via Group Codes," IEEE Transactions on Information Theory, Vol. IT-15, pp. 72-78, 1969.
2. T.R. Crimmins, and H.M. Horwitz, "Mean-Square Error Optimum Coset Leaders for Group Codes," IEEE Transactions on Information Theory, Vol. IT-16, pp. 429-432, 1970.
3. G. Bachman, Elements of Abstract Harmonic Analysis. New York: Academic Press, 1964.
4. W. Rudin, Fourier Analysis on Groups. New York: Interscience, 1967.
5. L.H. Loomis, Harmonic Analysis. New York: Mathematical Association of America, 1965.
6. L.S. Pontryagin, Topological Groups. New York: Gordon and Breach, 1956.
7. N.J. Fine, "The Generalized Walsh Functions," Transactions of the American Mathematical Society, Vol. 69, pp. 66-77, 1950.

Appendix

This appendix contains the details of the proofs of the results contained in the body of this paper.

Proof of Lemma.

Suppose that u and v are in V . We wish to show that $\lambda(u+v) = \lambda(u) \oplus \lambda(v)$. Towards this end we will need a basis for V . But since V^* is isomorphic with V , corresponding to the basis $\{x_i\}_{i=1}^k$ of V^* is the basis $\{e_i\}_{i=1}^n$ of V . Furthermore $\{e_i\}_{i=1}^k$ may be taken as a basis for G while $\{e_i\}_{i=k+1}^n$ completes the basis of V . Specifically we associate the basis with the given characters by:

$$x_i(v) = \begin{cases} -1 & \text{if } v_i = -1 \\ +1 & \text{if } v_i = 0 \end{cases} \quad \text{where } v = v_1 e_1 + \dots + v_n e_n \quad (A1)$$

$i=1, 2, \dots, n$

Let

$$u = \sum_{i=1}^n b_i e_i \quad (A2a)$$

$$v = \sum_{i=1}^n c_i e_i \quad (A2b)$$

Thus

$$\lambda(u) = \sum_{i=1}^k 2^{k-i} \left[\frac{1 - (-1)^{b_i} \prod_{j=1}^n (-1)^{b_j a_{ij}}}{2} \right] \quad (A3a)$$

$$\lambda(v) = \sum_{i=1}^k 2^{k-i} \left[\frac{1 - (-1)^{c_i} \prod_{j=1}^n (-1)^{c_j a_{ij}}}{2} \right] \quad (A3b)$$

Also we have

$$\lambda(u+v) = \sum_{i=1}^k 2^{k-i} \left[\frac{1 - (-1)^{b_i+c_i} \prod_{j=1}^n (-1)^{b_j a_{ij} + c_j a_{ij}}}{2} \right] \quad (A3c)$$

So it remains to show for all $i=1, \dots, k$ that

$$\left[\frac{1 - (-1)^{b_i} \prod_{j=k+1}^n b_j a_{ij}}{2} \right] \oplus \left[\frac{1 - (-1)^{c_i} \prod_{j=k+1}^n c_j a_{ij}}{2} \right] = \left[\frac{1 - (-1)^{b_i+c_i} \prod_{j=k+1}^n (b_j+c_j) a_{ij}}{2} \right] \quad (A4)$$

Let $m=n-k$ and assume that $\sum_{j=k+1}^n$ means 0. We will demonstrate the above by induction on m . Suppose $m=0$. Then by means of a truth table we have:

$$\left[\frac{1 - (-1)^{b_i}}{2} \right] \oplus \left[\frac{1 - (-1)^{c_i}}{2} \right] = \left[\frac{1 - (-1)^{b_i+c_i}}{2} \right] \quad (A5)$$

Now suppose that (A4) is true for $m=n-k-2$. The same truth table shows that the following is also true.

$$\left[\frac{1 - (-1)^{y_i} \prod_{j=1}^n b_j a_{ij}}{2} \right] \oplus \left[\frac{1 - (-1)^{z_i} \prod_{j=1}^n c_j a_{ij}}{2} \right] = \left[\frac{1 - (-1)^{y_i+z_i} \prod_{j=1}^n (b_j+c_j) a_{ij}}{2} \right] \quad (A6)$$

Therefore (A4) is valid for $m=n-k+1$. Take any $u \in S$. We may express this as:

$$u = r_1 2^0 + r_2 2^1 + \dots + r_k 2^{k-1} \quad (A7)$$

It is easy to show that if we specify $u \in V$ by

$$u = v_1 e_1 + \dots + v_k e_k \quad \text{where } (-1)^{v_i} = r_i, \quad i=1, 2, \dots, k. \quad (A8)$$

then $\lambda(u) = r$ and $u \in G$. Therefore λ is from G onto S . Since the sets S and G are finite and of equal size, the restriction λ_G is an isomorphism.

Suppose λ is a homomorphism whose restriction is an isomorphism. $\{2^0, 2^1, \dots, 2^{k-1}\}$ is a basis for S over the binary field. Hence there is basis for G , $\{e_i\}_{i=1}^k$, such that $\lambda_G(e_i) = 2^{k-i}$.

Define a set of characters as:

$$\chi_i(e_j) = (-1)^{\delta_{ij}} \quad i, j = 1, 2, \dots, k \quad (A9)$$

δ_{ij} is the Kronecker Delta

The set of characters in V/A^* , defined as $\{\chi_i\}_{i=1}^k$ is a basis for V/A^* .

The basis for G may be expanded into one for V by adding the vectors $\{e_j\}_{j=k+1}^n$. Define another set of characters $\{\chi_j\}_{j=k+1}^n$ in terms of them.

$$\chi_j(e_m) = (-1)^{\delta_{jm}} \quad j, m = k+1, \dots, n \quad (A10)$$

The χ_i 's are linearly independent and are constant on G ; so they are a basis for A^* . If a generic member of V is $v = \sum_{i=1}^n b_i e_i$, then since λ is a homomorphism:

$$\lambda(v) = \sum_{i=1}^n b_i \lambda(e_i) \quad (A11)$$

$$= \sum_{i=1}^k b_i 2^{k-i} \oplus \sum_{j=k+1}^n b_j \lambda(e_j) \quad (A12)$$

From a truth table it is obvious that

$$b_i = \left[\frac{1 - \chi_i(v)}{2} \right] \quad i = 1, 2, \dots, n \quad (A13)$$

Further each $\lambda(e_j)$ may be expressed:

$$\lambda(e_j) = \sum_{m=1}^k a_{mj} 2^{k-m} \quad j = k+1, \dots, n \quad (A14)$$

Thus combining (A12) and (A13) with (A14), we find

$$\lambda = \sum_{i=1}^k 2^{k-i} \left\{ \left[\frac{1 - \chi_i}{2} \right] \oplus \sum_{j=k+1}^n a_{ij} \left[\frac{1 - \chi_i}{2} \right] \right\} \quad (A15)$$

By an induction argument similar to the one given previously it is easy to show the following identity

$$\left[\frac{1 - \chi_i}{2} \right] \oplus \sum_{j=k+1}^n a_{ij} \left[\frac{1 - \chi_i}{2} \right] = \left[\frac{1 - \chi_i \sum_{j=k+1}^n \chi_j a_{ij}}{2} \right] \quad (A16)$$

The uniqueness for a given G follows from the linear independence of the members of V^* in the

space of square-summable functions defined on V .

Suppose λ is also a projection mapping taking the m basis elements of V , $\{v_1, \dots, v_m\}$, into zero. Since λ is an isomorphism on G , we may replace m of the basis elements labeled e_{k+1}, \dots, e_n above by v_1, v_2, \dots, v_m through use of the Exchange Theorem [6]. For the sake of definiteness we may assume that the first m e_i 's are replaced. Hence, since $\lambda(v_i) = 0$, we have from equation (A14) that $a_{ij} = 0$ for all $i = 1, 2, \dots, k$.

Now suppose that in the representation of λ by equation (4) that there are m indices, j_1, \dots, j_m where $k+1 \leq j_l \leq n, l = 1, 2, \dots, m$, such that $a_{ij_l} = 0$ for all $i = 1, 2, \dots, k$. Select the corresponding linearly independent characters χ_{j_l} and basis elements e_{j_l} associated with them according to equation (A1). It immediately follows from equation (A1) and equation (4) that $\lambda(e_{j_l}) = 0$ for any $l = 1, 2, \dots, m$.

Proof of Theorem:

The mean square error is given as

$$\begin{aligned} \epsilon^2 &= E\{(\lambda_G(g) - \lambda(v))^2\} \\ &= \sum_{g \in G} \sum_{v \in V} (\lambda_G(g) - \lambda(v))^2 P(g, v) \quad (A17) \\ &= \frac{1}{2^k} \sum_{g \in G} \sum_{v \in V} (\lambda_G(g) - \lambda(v))^2 P(v = g | 0) \end{aligned}$$

By expanding and collecting terms we must consider three terms. Since λ_G is an isomorphism, the first of these terms is:

$$\sum_{g \in G} \lambda_G^2(g) = \sum_{i=0}^{2^k-1} i^2 = \frac{2^k(2^k-1)(2^{k+1}-1)}{6} \quad (A18)$$

The second term is expressed using transforms.

$$\begin{aligned} &\sum_{g \in G} \sum_{v \in V} \lambda^2(v) P(v = g | 0) \\ &= \frac{1}{2^{3n}} \sum_{g \in G} \sum_{\alpha \in V^*} \sum_{\beta \in V^*} \sum_{\gamma \in V^*} \hat{\lambda}(\alpha) \hat{\lambda}(\beta) \hat{P}(\gamma) \gamma(-g) \sum_{v \in V} \alpha(v) \beta(v) \gamma(v) \quad (A19) \end{aligned}$$

The following identities are easy to show.

$$\sum_{v \in V} \alpha(v) \beta(v) \gamma(v) = \begin{cases} 2^n & \text{if } \alpha \beta \gamma = \chi_0 \\ 0 & \text{otherwise} \end{cases} \quad (A20)$$

(χ_0 is the identity in V^*)

$$\sum_{g \in G} \lambda(g) = \begin{cases} 2^k & \text{if } \lambda \in A^* \\ 0 & \text{if } \lambda \notin A^* \end{cases} \quad (A21)$$

So the second term becomes:

$$\sum_{g \in G} \sum_{v \in V} \lambda^2(v) P(v-g|0) =$$

$$\frac{2^k}{2^{2n}} \sum_{(\alpha\beta) \in A^*} \hat{\lambda}(\alpha) \hat{\lambda}(\beta) \hat{P}(\alpha\beta) \quad (A22)$$

The third term in the expansion of (A17) is:

$$\begin{aligned} & \sum_{g \in G} \sum_{v \in V} \lambda_G(g) \lambda(v) P(v-g|0) \\ &= \frac{1}{2^{3n}} \sum_{\alpha \in V^*} \sum_{\beta \in V^*} \sum_{\gamma \in V^*} \\ & \hat{\lambda}(\alpha) \hat{\lambda}(\beta) \hat{P}(\gamma) \left(\sum_{g \in G} \alpha(g) \gamma(g) \sum_{v \in V} \beta(v) \gamma(v) \right) \quad (A23) \\ &= \frac{1}{2^{2n-k}} \sum_{(\alpha\beta) \in A^*} \hat{\lambda}(\alpha) \hat{\lambda}(\beta) \hat{P}(\beta) \end{aligned}$$

Finally we have an expression for the mean-square error.

$$\begin{aligned} \epsilon^2 &= \frac{(2^k-1)(2^{k+1}-1)}{6} \\ &+ \frac{1}{2^{2n}} \left[\sum_{(\alpha\beta) \in A^*} \hat{\lambda}(\alpha) \hat{\lambda}(\beta) \hat{P}(\alpha\beta) \right] \quad (A24) \\ &- \frac{2}{2^{2n}} \sum_{(\alpha\beta) \in A^*} \hat{\lambda}(\alpha) \hat{\lambda}(\beta) \hat{P}(\beta) \end{aligned}$$

The summation indexed by $(\alpha\beta) \in A^*$ may be expressed in two parts.

$$\sum_{(\alpha\beta) \in A^*} = \sum_{\alpha \in A^*} \sum_{\beta \in A^*} + \sum_{\substack{\alpha \text{ and } \beta \\ \text{in same} \\ \text{coset of } A^*}} \quad (A25)$$

The lemma gives an expression for λ . It is easy to show that the transform λ of such a mapping has the following properties

$$\hat{\lambda}(\alpha) = \begin{cases} \sum_{v \in V} \lambda(v) = 2^{n-1}(2^k-1) & \text{if } \alpha = \chi_0 \\ -\left(\frac{2^{n+k}}{2}\right) \frac{1}{2^r} & \text{if } \alpha = \chi_r \prod_{j=k+1}^n \chi_j^{a_{1j}} \\ 0 & \text{otherwise} \end{cases} \quad (A26)$$

Combining equations (A25) and (A26) allows us to evaluate one of the terms in equation (A24)

$$\begin{aligned} & \sum_{(\alpha\beta) \in A^*} \hat{\lambda}(\alpha) \hat{\lambda}(\beta) \hat{P}(\alpha\beta) \\ &= (2^{n-1}(2^k-1))^2 + \frac{2^{2n-2}}{3} (2^{2k-1}) \quad (A27) \end{aligned}$$

Thus

$$\begin{aligned} \epsilon^2 &= \frac{(2^k-1)(2^{k+1}-1)}{3} \\ &- \frac{2}{2^{2n}} \sum_{(\alpha\beta) \in A^*} \hat{\lambda}(\alpha) \hat{\lambda}(\beta) \hat{P}(\beta) \quad (A28) \end{aligned}$$

Also using (A26)

$$\begin{aligned} & \sum_{(\alpha\beta) \in A^*} \hat{\lambda}(\alpha) \hat{\lambda}(\beta) \hat{P}(\beta) = 2^{2n-2} (2^k-1)^2 \hat{P}(\chi_0) \\ &+ 2^{2n+2k-2} \sum_{r=1}^k \frac{1}{4^r} \hat{P}(\chi_r \prod_{j=k+1}^n \chi_j^{a_{rj}}) \quad (A29) \end{aligned}$$

So the mean-square error becomes:

$$\epsilon^2 = \frac{4^k-1}{6} - 2^{2k-1} \sum_{r=1}^k \frac{1}{4^r} \hat{P}(\chi_r \prod_{j=k+1}^n \chi_j^{a_{rj}}) \quad (A30)$$

Suppose that λ is an optimum mapping of the required type and that G is the subgroup, A is its annihilator, and that the independent characters $\{\chi_i\}_i^n$ and the constants a_{ij} are its characterization in terms of the lemma. Specifically let $\{\chi_j\}_{j=k+1}^n$ be the basis for A^* and

$\chi_1 A^*, \dots, \chi_k A^*$ be the one for V^*/A^* . Now suppose that ζ is any other mapping of the same type but with subgroup H , and its annihilator B^* . Let the basis for B^* be denoted as $\phi_{k+1}, \dots, \phi_n$ while the basis for V^*/B^* be denoted as $\phi_1 B^*, \phi_2 B^*, \dots, \phi_k B^*$. Further let the b_{ij} 's be the constants in the formula for the characterization of ζ .

Because of the optimality assumptions we find a necessary condition to be:

$$\begin{aligned} & \sum_{r=1}^k \frac{1}{4^r} \hat{P}(\chi_r \prod_{j=k+1}^n \chi_j^{a_{rj}}) \geq \\ & \sum_{r=1}^k \frac{1}{4^r} \hat{P}(\phi_r \prod_{j=k+1}^n \phi_j^{b_{rj}}) \quad (A31) \end{aligned}$$

A fact that we will use shortly is:

$$\begin{aligned} & \hat{P}(\chi_r \prod_{j=k+1}^n \chi_j^{a_{rj}}) \geq \\ & \hat{P}(\chi_{i+1} \prod_{j=k+1}^n \chi_j^{a_{i+1,j}}), i=1, 2, \dots, k-1 \quad (A32) \end{aligned}$$

For we may make the following choice for the other mapping rule ζ . $\phi_j = \chi_j$ for all $j = 1, 2, \dots, n$, except $\phi_{i+1} = \chi_i$ and $\phi_i = \chi_{i+1}$.

Also take all $a_{rj} = b_{rj}$ except $b_{ij} = a_{i+1,j}$ for all j and $b_{i+1,j} = a_{ij}$ for all j . Then since λ is optimal we must have by inequality (A31):

$$\begin{aligned} & \frac{1}{4^i} \hat{P}(\chi_i \prod_{j=k+1}^n \chi_j^{a_{ij}}) + \frac{1}{4^{i+1}} \hat{P}(\chi_{i+1} \prod_{j=k+1}^n \chi_j^{a_{i+1,j}}) \\ & \geq \frac{1}{4^i} \hat{P}(\chi_{i+1} \prod_{j=k+1}^n \chi_j^{a_{i+1,j}}) + \frac{1}{4^{i+1}} \hat{P}(\chi_i \prod_{j=k+1}^n \chi_j^{a_{ij}}) \quad (A33) \end{aligned}$$

It also easily follows from the obvious choices of elements in inequality (A31) that

$$\hat{P}(x_{k+1} \prod_{j=k+1}^n x_j^{a_{kj}}) \geq \hat{P}(x_s), s=k+1, \dots, n \quad (A34)$$

Another consequence of optimality is:

$$\hat{P}(\psi_i) \geq \hat{P}(\beta) \text{ for all } \beta \in [\psi_1, \psi_2, \dots, \psi_{i-1}, x_{k+1}, \dots, x_n]. \quad (A35)$$

where $\psi_i = x_1 \prod_{j=k+1}^n x_j^{a_{ij}}$ with $1 \leq i \leq k$

To demonstrate this we first note that there must be an m , with $i \leq m \leq k$ such that $\{\psi_1, \dots, \psi_{m-1}, \beta, \psi_{m+1}, \dots, \psi_k\}$ is a linearly independent set by virtue of the Exchange Theorem [6]. Next we construct another mapping rule ζ by choosing $\phi_j = x_j$ for all j except m and in

that case we set $\phi_m = \beta \prod_{j=k+1}^n x_j^{a_{mj}}$. Also we take binary constants $a_{rj} = b_{rj}$ for all r and j indices in the proper sets. Then after cancellation we have:

$$\frac{1}{4^m} \hat{P}(x_m \prod_{j=k+1}^n x_j^{a_{mj}}) \geq \frac{1}{4^m} \hat{P}(\beta) \quad (A36)$$

But since m is between i and k , an application of the inequality (A32) completes the subproof of inequality (A35).

If $\beta \in [x_{k+1}, \dots, x_n]$, there is some ℓ , $k+1 \leq \ell \leq n$ such that $\{x_{k+1}, \dots, x_{\ell-1}, \beta, x_{\ell+1}, \dots, x_n\}$ is equivalent to the original set of linearly independent characters.

By choosing $\phi_k = \beta$, and $b_{kj} = 0$ for $j = k+1, \dots, n$ but all $i \neq k$, taking $\phi_i = x_i$ and $a_{ij} = b_{ij}$ we have from inequality (A31):

$$\hat{P}(x_{k+1} \prod_{j=k+1}^n x_j^{a_{kj}}) \geq \hat{P}(\beta) \text{ for all } \beta \in [x_{k+1}, \dots, x_n] \quad (A37)$$

Coupling this result with inequality (A32) we establish the following strengthened version of inequality (A35).

$$\hat{P}(\psi_i) \geq \hat{P}(\beta) \text{ for all } \beta \in [\psi_1, \dots, \psi_{i-1}] \quad (A38)$$

where $\psi_i = x_1 \prod_{j=k+1}^n x_j^{a_{ij}}$ with $i=1, 2, \dots, k$.

Since the $\{\psi_i\}_{i=1}^k$ as defined above are also linearly independent characters, we may relabel them so as to obtain part ii of the theorem.

An application of the lemma gives equation (6) while equation (5) follows from the duality of subgroups and quotient groups [6].

Suppose that the rule λ is determined by inequality (7). Further suppose that ξ is another homomorphism of the required type and that its characterization by the lemma is in terms of linearly independent characters

$\{\phi_i\}_{i=1}^n$, binary constants b_{ij} , a annihilator B^* and corresponding subgroup H . From equation (A30) it suffices to show that

$$\sum_{r=1}^k \frac{1}{4^r} [\hat{P}(x_r) - \hat{P}(\phi_r \prod_{j=k+1}^n \phi_j^{b_{rj}})] \geq 0 \quad (A39)$$

But each of the linearly independent characters x_r for $r = 1, 2, \dots, k$ were chosen in order and in a maximal fashion according to inequality (7). Therefore each term in the sum of the desired inequality (A39) is nonnegative and thus the inequality is true. Hence we have demonstrated the sufficiency part of the theorem as well.

ON ORDERING OF A CLASS OF GENERALIZED WALSH FUNCTIONS

by

Sze-Hou Chang

Department of Electrical Engineering
Northeastern University, Boston, Massachusetts 02115
and

Thomas Joseph

Department of Mathematics
Northeastern University, Boston, Massachusetts 02115

Abstract

H.E. Chrestenson (4) defines a class of Generalized Walsh functions using the concept of Generalized Rademacher functions. But this is ordered neither according to symmetry nor according to sequency. Here a generalized concept of symmetry is introduced and an ordering of the above class of generalized Walsh functions is presented in terms of it. A measure of oriented phase shift, which is a generalized concept of sequency, is introduced and an ordering in terms of this is given. Two difference equations satisfied by this class of generalized Walsh functions, one ordering according to symmetry and the other ordering according to sequency are presented. The symmetry index leads naturally to the tree graph as well as a group structure.

Introduction

The Walsh functions form a complete set of orthonormal sequences with elements $\{1, -1\}$. Higher order Walsh functions are most conveniently generated by successive tensor products of the first order Walsh functions. This process is further simplified by mapping the elements $\{1, -1\}$ to $\{0, 1\}$ and replacing tensor products by tensor sums. It is known that there are 2^k Walsh functions of order k which may be ordered according to sequency, a term derived from frequency signifying the number of zero-crossings in the waveform representing the functions or sequence (8). Also it is known that the 2^k sequences can be generated and arranged in terms of odd and even symmetry (9). A difference equation defining the Walsh functions in an ordering according to sequency is given by Harmuth (8).

Generalized Hadamard Matrices

A Hadamard matrix is an orthogonal square matrix whose elements are $+1$ and -1 . Two types of generalizations of Hadamard matrices have been studied. (i) Orthogonal square matrix $H(\ell, h)$ of order h with $\ell > 2$ distinct integers as elements (7, 10). (ii) Orthogonal square matrix $H(\ell, h)$ of order h all of whose elements are ℓ th roots of unity (5). A subset of the latter type of generalized Hadamard matrices forms a class of generalized Walsh functions

which was defined by Chrestenson as products of generalized Rademacher functions.

In the generalized Walsh functions the binary elements $\{0, 1\}$ are extended to ℓ -ary (ℓ an integer) elements $i = \{0, 1, 2, \dots, \ell - 1\}$, considered as the phase angles in units of $\frac{2\pi}{\ell}$, which are mapped to the ℓ th roots of unity:

$$i \rightarrow e^{(j2\pi/\ell)i} = \gamma^i.$$

The orthogonality and completeness are preserved in the extension (4).

Tensor Product and Tensor Sum

The generalized Walsh functions of order 1 and k are expressible as generalized Hadamard matrices as follows,

$$H^{(1)} = \begin{bmatrix} \gamma^0 & \gamma^0 & \gamma^0 & \dots & \gamma^0 \\ \gamma^0 & \gamma^1 & \gamma^2 & \dots & \gamma^{\ell-1} \\ \gamma^0 & \gamma^2 & \gamma^4 & \dots & \gamma^{\ell-2} \\ \dots & \dots & \dots & \dots & \dots \\ \gamma^0 & \gamma^{\ell-1} & \gamma^{\ell-2} & \dots & \gamma^1 \end{bmatrix}$$

$$H^{(k)} = H^{(1)} \otimes H^{(1)} \otimes H^{(1)} \dots \otimes H^{(1)}.$$

where the tensor product of two matrices A and B is given by

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & \dots & a_{2n}B \\ \dots & \dots & \dots & \dots \\ a_{n1}B & a_{n2}B & \dots & a_{nn}B \end{bmatrix}$$

Both of these matrices $H^{(1)}$ and $H^{(k)}$ are orthonormal over the field of complex numbers. These matrices can be obtained from the map $i \rightarrow \gamma^i$ of each element of the matrices $h^{(1)}$ and $h^{(k)}$ given by

$$h^{(1)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 2 & 3 & 4 & \dots & \ell-1 \\ 0 & 2 & 4 & 6 & 8 & \dots & \ell-2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \dots & \cdot \\ 0 & \ell-1 & \ell-2 & \cdot & \cdot & \dots & 1 \end{bmatrix}$$

$$h^{(k)} = h^{(1)}[+] h^{(1)}[+] h^{(1)}[+] \dots [+] h^{(1)}$$

where the tensor sum of two matrices A and B is given by

$$A[+]B = \begin{bmatrix} a_{11}+B & a_{12}+B & \dots & a_{1n}+B \\ a_{21}+B & a_{22}+B & \dots & a_{2n}+B \\ \cdot & \cdot & \dots & \cdot \\ a_{n1}+B & a_{n2}+B & \dots & a_{nn}+B \end{bmatrix}$$

where the sum $a + B$ indicates the matrix obtained by adding a to every element of B , the addition being taken modulo ℓ . It is to be noted that $h^{(k)}$ is no longer orthonormal. It represents a vector space of dimension k over the ring of integers modulo ℓ (according to the definition of vector space given in Van Der Waerden (11)).

Symmetry Index for Generalized Walsh Functions

The ℓ rows of the matrix $h^{(1)}$ define ℓ different symmetries among the components of the row vectors, called " ℓ -ary symmetries". Equivalently in the mapped case the corresponding rows of $H(1)$ define ℓ different " ℓ -ary symmetries". For example the ternary symmetries are given by the rows of $h^{(1)}$ when $\ell = 3$ namely, $(0 \ 0 \ 0)$, $(0 \ 1 \ 2)$ and $(0 \ 2 \ 1)$ or equivalently for $H(1)$ $(\gamma^0 \ \gamma^0 \ \gamma^0)$ or equivalently $(\gamma^0 \ \gamma^1 \ \gamma^2)$ and $(\gamma^0 \ \gamma^2 \ \gamma^1)$ which can be represented as in Fig. i.

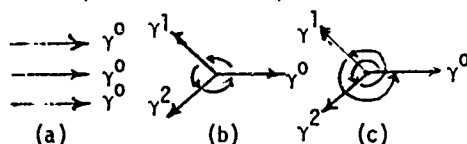


Fig. i. Ternary symmetries (a) with index 0; (b) with index 1; (c) with index 2.

This concept of ternary symmetry was used in representing 3-phase unbalanced power systems by symmetrical components. Any cyclic permutation of a symmetry is considered as belonging to the same symmetry with a certain initial phase. Thus $(0 \ 1 \ 2)$ and $(1 \ 2 \ 0)$ are the same symmetry with initial phase 0 and 1 respectively.

If we denote $U_1 = (0 \ 1 \ 2 \ \dots \ \ell-1)$, the ℓ -ary symmetry of index α is given by

$$\alpha U_1 = (0 \ \alpha \ \alpha+1 \ \dots \ -\alpha) \text{ where}$$

$$\alpha = 0, 1, \dots, \ell-1.$$

A set of k basis vectors for $h^{(k)}$ can be obtained in terms of the ℓ -component vectors $U_0 = (000 \ \dots \ 0)$ and $U_1 = (012 \ \dots \ \ell-1)$ as follows

$$U_{100} \ \dots = U_1[+] U_0[+] U_0[+] \dots [+] U_0$$

$$U_{010} \ \dots = U_0[+] U_1[+] U_0[+] \dots [+] U_0$$

$$\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot$$

$$U_{000} \ \dots = U_0[+] U_0[+] U_0[+] \dots [+] U_1$$

The general row of $h^{(k)}$ considered as a vector with index $\alpha = (\alpha_0 \alpha_1 \ \dots \ \alpha_{k-1})$, referred as subscript, is obtained by linear combination of the basis vectors. Each basis vector is multiplied by the corresponding component of the index and the addition is digit by digit modulo ℓ . Consider an element $U_{\alpha_0 \alpha_1 \ \dots \ \alpha_{k-1}}$ belonging to $h^{(k)}$.

This vector has ℓ^k digits. The symmetry index $\alpha_0 \alpha_1 \ \dots \ \alpha_{k-1}$ implies that the digits are related by symmetry α_{k-1} (or its cyclic permutation) inside each subblock of ℓ digits, and the subblocks of ℓ digits are related by symmetry α_{k-2} (or its cyclic permutation) inside each block of ℓ^2 digits and so on. Lastly, all the ℓ^k digits are related in blocks of ℓ^{k-1} by symmetry α_0 . Thus given the symmetry index, a vector can be written without referring to any other member of the set $h^{(k)}$. Also the ℓ -ary symmetry of the Walsh functions shows that they can be generated as a tree code. This is illustrated by examples 1 and 2.

Example 1

Consider $\ell = 3$, $k = 2$.

$$h^{(1)} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 2 & 1 \end{bmatrix}$$

$$h^{(2)} = h^{(1)}[+] h^{(1)} = \begin{bmatrix} 0+h^{(1)} & 0+h^{(1)} & 0+h^{(1)} \\ 0+h^{(1)} & 1+h^{(1)} & 2+h^{(1)} \\ 0+h^{(1)} & 2+h^{(1)} & 1+h^{(1)} \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 \\ 0 & 2 & 1 & 0 & 2 & 1 & 0 & 2 \\ 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 \\ 0 & 1 & 2 & 1 & 2 & 0 & 2 & 0 \\ 0 & 2 & 1 & 1 & 0 & 2 & 2 & 1 \\ 0 & 0 & 0 & 2 & 2 & 2 & 1 & 1 \\ 0 & 1 & 2 & 2 & 0 & 1 & 1 & 2 \\ 0 & 2 & 1 & 2 & 1 & 0 & 1 & 2 \end{bmatrix}$$

The basis vectors for $h^{(2)}$ are

$$U_{10} = U_1 [+] U_0 = (0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 2 \ 2 \ 2)$$

$$U_{01} = U_0 [+] U_1 = (0 \ 1 \ 2 \ 0 \ 1 \ 2 \ 0 \ 1 \ 2)$$

A vector of symmetry index, say, 21 is obtained

$$U_{21} = 2U_{10} + U_{01} = (0 \ 1 \ 2 \ 2 \ 0 \ 1 \ 1 \ 2 \ 0)$$

where the addition is digit by digit modulo 3. Clearly the symmetry inside every subblock of 3 digits is of type 1 and the symmetry between the 3 subblocks is of type 2. Representing as a tree code, Fig. ii, we note that the first branching is determined by the symmetry index α_0 and the next branching is specified by the symmetry index $\alpha_0 \alpha_1$, and so on.

Index Vector	Code Vector
$\alpha_0 \alpha_1 \alpha_2$	
0 0	0 0 0 0 0 0
1 0	0 0 1 1 1 2 2 2
2 0	2 2 2 1 1 1
0 1	0 1 2 0 1 2
1 1	0 1 2 1 2 0 2 0
2 1	2 0 1 1 2 0
0 2	0 2 1 0 2 1
1 2	2 1 1 0 2 2 1 0
2 2	2 1 0 1 0 2

Fig. ii. Tree code representation of the 3^2 pre-mapped Walsh functions.

Example 2. Consider $\ell = 4$, $k = 2$, then

$$h^{(1)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 \\ 0 & 2 & 0 & 2 \\ 0 & 3 & 2 & 1 \end{bmatrix}$$

A set of basis vectors of $h^{(2)}$ for symmetry indexing is given by

$$U_{10} = U_1 [+] U_0 = 0000111122223333$$

$$U_{01} = U_0 [+] U_1 = 0123012301230123$$

A vector of symmetry index, say, 32 is obtained thus

$$U_{32} = 3U_{10} + 2U_{01} = 0202313120201313.$$

Clearly the symmetry inside every subblock of 4 digits is of type 2 and the symmetry between the 4 subblocks is of type 3. Tree code representation of $h^{(2)}$ is given in Fig. iii.

Index	Code Vectors
$\alpha_0 \alpha_1 \alpha_2$	
0 0	000000000000
1 0	111122223333
2 0	222200002222
3 0	333322221111
0 1	012301230123
1 1	123023013012
2 1	1 2 3 230101232301
3 1	301223011230
0 2	0 0 2 0202020202
1 2	131320203131
2 2	2 0 2 202002022020
3 2	313120201313
0 3	032103210321
1 3	103221033210
2 3	3 2 1 210303212103
3 3	321021031032

Fig. iii. Tree code formed by the 4^2 pre-mapped Walsh functions.

Phase Index and Generalized Sequence

Selecting the basis vectors properly, as given below, the 2^k vectors can be ordered in terms of phase number or sequence number β ranging from 0 to $2^k - 1$ in the increment of 1. It can be shown that this number β is equal to the amount of oriented angular phase shifts in the units of $2\pi(\ell - 1)/\ell$ of the vector $V_\beta = V \beta_0 \beta_1 \dots \beta_{k-1}$ where $\beta_i = \beta_0 \beta_1 \dots \beta_{k-1}$ is called the phase or sequence index and

$$\beta = \sum_{i=0}^{k-1} \beta_i \ell^i \quad \text{the phase or sequence number.}$$

$$V_1 = (0 \ 1 \ 2 \ \dots \ \ell - 1)$$
 as follows:

$$v_{0100 \dots 0} = (l-1)v_1[+] v_1[+] v_0[+] \dots [+]v_0$$

• • • , • • • • •

A general row of $h^{(k)}$ considered as a vector with index $\underline{\beta} = \beta_0 \beta_1 \dots \beta_{k-1}$ is obtained by the linear combination of the above basis vectors, thus

$$+ \beta_1 v_{0100 \dots 0} + \dots + \beta_{k-1} v_{0000 \dots 1}$$

$$\beta_i = \{0, 1, 2, \dots, l-1\},$$

$$i = \{0, 1, \dots, k - 1\}.$$

$$V_{100} = V_1 [+]\ V_0 [+]\ V_0$$

$$V_{010} = 2V_1 \quad [+] \quad V_1 \quad [+] \quad V_0$$

$$= \underset{(1)}{0} \underset{(1)}{0} \underset{(1)}{0} \underset{(1)}{1} \underset{(1)}{1} \underset{(1)}{2} \underset{(1)}{2} \underset{(1)}{2} \underset{(1)}{2} \underset{(1)}{2} \underset{(1)}{2} \underset{(1)}{0} \underset{(1)}{0} \underset{(1)}{0} \underset{(1)}{1} \underset{(1)}{1} \underset{(1)}{1} \underset{(1)}{1} \underset{(1)}{1} \underset{(1)}{2} \underset{(1)}{2} \underset{(1)}{2} \underset{(1)}{0} \underset{(1)}{0} \underset{(1)}{0}$$

$$V_{G01} = V_G [+]\ 2V_1 [+]\ V_1$$

[illegible]

$$V_{211} = 2V_{100} + 1V_{010} + 1V_{001}$$

$$= \begin{matrix} 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 \\ (1)(1)(1)(1)(1)(1)(1)(1)(1)(2)(1)(1)(1)(1)(1)(1)(1)(2)(1)(1)(1)(1)(1)(1)(1)(1) \end{matrix}$$

that one can be obtained from the other very easily. Let the symmetry index and the phase index of an element of the set of 2^k vectors be $\underline{\alpha} = \alpha_0 \alpha_1 \dots \alpha_{k-1}$ and $\underline{\beta} = \beta_0 \beta_1 \dots \beta_{k-1}$ respectively. Then they are related by

$$\beta_i = \sum_{j=i}^{k-1} \alpha_j, \quad i = 0, 1, 2, \dots, k-1$$

(*) or

$$\alpha_j = \beta_j - \beta_{j+1} \quad \text{and} \quad \alpha_{k-1} = \beta_{k-1}$$

$$j = 0, 1, \dots, k - 2.$$

Thus when the sequence number of a waveform belonging to the set of 2^k vectors is given, the corresponding sequence index and symmetry index can be obtained and the vector can be written from left to right using symmetry index without referring to any other member of

340

the set. The symmetry index of the tensor sum of two vectors with symmetry indices is the direct sum of the separate indices. This along with the relations (*) is useful in obtaining sequency indices and sequency numbers of higher order vectors from lower order ones in an iterative form.

Symmetry Index and Period

Let U_{α} , an element of $h^{(k)}$ with symmetry index $\alpha = \alpha_0 \alpha_1 \dots \alpha_{k-1}$, be represented by

$$U_{\alpha} = (a_0, a_1, \dots, a_i, \dots, a_{\ell^k - 1})$$

where

$$a_i = \{0, 1, \dots, \ell - 1\}.$$

Then the period $p^{(k)}$ of the corresponding ℓ^k component discrete Walsh function is the smallest number for which $a_{m+p^{(k)}} = a_m$ where the addition is modulo (ℓ^k) . Let $\underline{p}^{(k)} = p_0 p_1 \dots p_k$ be the ℓ -ary expression of $p^{(k)}$ where

$$p^{(k)} = \sum_{i=0}^k p_i \ell^i.$$

that is, corresponding to $\alpha = \alpha_0 \alpha_1 \dots \alpha_{k-1}$ we have $\underline{p}^{(k)} = p_0 p_1 \dots p_k$. The following result shows that the period $p^{(k)}$ can be obtained from the symmetry index. The converse is not true as there is no one-to-one correspondence between period and symmetry index.

(i) ℓ is a prime number

$$p_{k-i} = \begin{cases} 0 & \text{if } \alpha_i = 0 \\ 1 & \text{if } \alpha_i \neq 0 \end{cases}$$

and $p_i = 0$, if $p_{i+1} \neq 0$, $i = 0, 1, \dots, k-1$,

and $p_0 = 1$ if $p_1 = p_2 = \dots = p_k = 0$.

(ii) ℓ is not a prime number

$$(\alpha_i, \ell) = 1$$

$$p_{k-i}$$

and

$$p_{k-i-1} = \begin{cases} n & \text{if } \alpha_i \neq \ell, \quad n = \frac{\ell}{\alpha_i} \\ m & \text{if } \alpha_i = \ell, \quad m = \frac{\ell}{(\alpha_i, \ell)} \end{cases}$$

and

$$p_i = 0, \text{ if } p_{i+1} \neq 0, i = 0, 1, \dots, k-1,$$

and

$$p_0 = 1 \text{ if } p_1 = p_2 = \dots = p_k = 0,$$

where (x, y) is the greatest common divisor of x and y . See example 4. For the binary case, a similar result has been reported (12).

Difference Equations

Generalized Walsh functions can be defined by either of the following difference equations.

(a) A difference equation which orders the Walsh functions in terms of a number called symmetry number which we denote by α . As is given in example 4, representing the set of ℓ^k Walsh functions of order k as ℓ^k -component vectors, the vector $W(\alpha, \theta)$ has symmetry number α and symmetry index $\alpha_0 \alpha_1 \dots \alpha_{k-1}$ where

$$\alpha = \sum_{i=0}^{k-1} \alpha_i \ell^i.$$

We write the difference equation for the generalized Walsh functions in the following form:

$$W(\ell n + q, \theta) = W(n, \ell \theta) + \gamma^0 W(n, \ell \theta - 1) + \dots + \gamma^{(\ell-1)q} W(n, \ell \theta - \ell + 1)$$

where

$$n = 0, 1, 2, \dots, \ell^{k-1} - 1;$$

$$q = 0, 1, 2, \dots, \ell - 1;$$

$$\gamma = e^{2j\pi/\ell}; \quad W(0, \theta) = \gamma^0, \quad 0 \leq \theta < 1;$$

and the exponent of γ is taken modulo ℓ .

See Example 4 on next page.

Example 4. Let $\ell = 3$ and $k = 2$

n	q	$\alpha_0 \alpha_1$	$W(\alpha, \theta)$	$W(\alpha, \theta)$ as 3^2 -component vector	Period	$p_0 p_1 p_2$
0	0	0 0	$W(0, \theta)$	$\gamma^0 \gamma^0 \gamma^0 \gamma^0 \gamma^0 \gamma^0 \gamma^0 \gamma^0$	1	1 0 0
0	1	1 0	$W(1, \theta)$	$\gamma^0 \gamma^0 \gamma^0 \gamma^1 \gamma^1 \gamma^1 \gamma^2 \gamma^2$	9	0 0 1
0	2	2 0	$W(2, \theta)$	$\gamma^0 \gamma^0 \gamma^0 \gamma^2 \gamma^2 \gamma^2 \gamma^1 \gamma^1$	9	0 0 1
1	0	0 1	$W(3, \theta)$	$\gamma^0 \gamma^1 \gamma^2 \gamma^0 \gamma^1 \gamma^2 \gamma^0 \gamma^1$	3	0 1 0
1	1	1 1	$W(4, \theta)$	$\gamma^0 \gamma^1 \gamma^2 \gamma^1 \gamma^2 \gamma^0 \gamma^2 \gamma^0$	9	0 0 1
1	2	2 1	$W(5, \theta)$	$\gamma^0 \gamma^1 \gamma^2 \gamma^2 \gamma^0 \gamma^1 \gamma^1 \gamma^2$	9	0 0 1
2	0	0 2	$W(6, \theta)$	$\gamma^0 \gamma^2 \gamma^1 \gamma^0 \gamma^2 \gamma^1 \gamma^0 \gamma^2$	3	0 1 0
2	1	1 2	$W(7, \theta)$	$\gamma^0 \gamma^2 \gamma^1 \gamma^1 \gamma^0 \gamma^2 \gamma^2 \gamma^1$	9	0 0 1
2	2	2 2	$W(8, \theta)$	$\gamma^0 \gamma^2 \gamma^1 \gamma^2 \gamma^1 \gamma^0 \gamma^1 \gamma^2$	9	0 0 1

(b) Walsh functions ordered according to sequence number β may be defined in a similar manner as in (a) by the following difference equation

$$W(\ell n + q, \theta) = W(n, \ell \theta) + \gamma^{q+m} W(n, \ell \theta - 1) + \dots + \gamma^{(\ell-1)(q+m)} W(n, \ell \theta - \ell - 1)$$

where $m = (\ell - 1) \frac{\ell n + q}{\ell}$; $[x]$ denotes the greatest integer less than or equal to x , $W(0, \theta) = \gamma^0$, $0 \leq \theta < 1$; and the exponent of γ is taken modulo ℓ .

Example 5 Let $\ell = 3$ and $k = 2$.

n	q	m	$W(\beta, \theta)$	$W(\beta, \theta)$ as 3^2 -component vector
0	0	0	$W(0, \theta)$	$\gamma^0 \gamma^0 \gamma^0 \gamma^0 \gamma^0 \gamma^0 \gamma^0 \gamma^0$
0	1	0	$W(1, \theta)$	$\gamma^0 \gamma^0 \gamma^0 \gamma^1 \gamma^1 \gamma^1 \gamma^2 \gamma^2$
0	2	0	$W(2, \theta)$	$\gamma^0 \gamma^0 \gamma^0 \gamma^2 \gamma^2 \gamma^2 \gamma^1 \gamma^1$
1	0	2	$W(3, \theta)$	$\gamma^0 \gamma^1 \gamma^2 \gamma^2 \gamma^0 \gamma^1 \gamma^1 \gamma^2$
1	1	2	$W(4, \theta)$	$\gamma^0 \gamma^1 \gamma^2 \gamma^0 \gamma^1 \gamma^2 \gamma^0 \gamma^1$
1	2	2	$W(5, \theta)$	$\gamma^0 \gamma^1 \gamma^2 \gamma^1 \gamma^2 \gamma^0 \gamma^2 \gamma^0$
2	0	4	$W(6, \theta)$	$\gamma^0 \gamma^2 \gamma^1 \gamma^1 \gamma^0 \gamma^2 \gamma^2 \gamma^1$
2	1	4	$W(7, \theta)$	$\gamma^0 \gamma^2 \gamma^1 \gamma^2 \gamma^1 \gamma^0 \gamma^1 \gamma^2$
2	2	4	$W(8, \theta)$	$\gamma^0 \gamma^2 \gamma^1 \gamma^0 \gamma^2 \gamma^1 \gamma^0 \gamma^2$

Discussions

The ordering of the class of generalized Walsh functions in terms of their symmetry index α or phase index β is pertinent in the following applications.

(a) In the generation and detection of Walsh waveforms, the implementation is simplified by the structure of indexing.

(b) The selection of appropriate Walsh functions for the analysis and synthesis of signal waveforms may be facilitated if the waveform can be attributed to a set of salient components with specified symmetries or phase shifts.

(c) Combination and decombination of Walsh functions of various orders are much facilitated by means of the indices.

(d) The entire set of 2^k Walsh functions form a group under digit by digit multiplication modulo 2. This group is represented by the group of symmetry indices under digit by digit addition modulo 2. In addition to the examples cited before, let's consider the case that 2 is a composite number, (examples 6 and 7) then the numbers modulo 2 can be listed as subgroups together with their cosets. This is also true for the number 2^k . The listing of Walsh functions in subgroups and cosets leads to further simplifications and insights.

Example 6 Let $2 = 6$, $k = 1$

Subgroup	0	2	4
Coset	1	3	5

In other words, since $2 = 3 \times 2$, a composite number, the entire set of 6 Walsh functions may be listed as a subgroup of 3 functions (possessing 3 different symmetries) plus a coset. It is also possible to list them as a subgroup of order 2 and 2 cosets.

Subgroup	0	3
Coset	1	4
Coset	2	5

Example 7 Let $2 = 6$, $k = 2$.

Subgroup	00	02	04	20	22	24	40	42	44
Coset	01	03	05	21	23	25	41	43	45
"	10	12	14	30	32	34	50	52	54
"	11	13	15	31	33	35	51	53	55

Here $2^k = 6^2 = 3^2 \cdot 2^2$, the entire set is listed as a subgroup of order 3^2 plus $(2^2 - 1) = 3$

cosets. The group operation is digit by digit addition modulo 6. Note that the listing of example 7 is a representation by symmetry indices of the following tensor sums

$$\begin{bmatrix} u_0 & u_2 & u_4 \\ u_1 & u_3 & u_5 \end{bmatrix} [+] \begin{bmatrix} u_0 & u_2 & u_4 \\ u_1 & u_3 & u_5 \end{bmatrix}$$

While this paper emphasizes primarily in introducing two indices α and β , it also serves the purpose of obtaining period p in terms of α and of relating three recurrence methods (i) difference equation (ii) tensor product and (iii) tensor sum, the last being easily handled especially in terms of the indices.

References

- (1) J.L. Walsh "A Closed Set of Orthogonal Functions" Am. Jour. of Math., Vol.45, pp. 5-24 (1923).
- (2) Proc. Symp. "Applications of Walsh Functions" March-April 1970 Symp. held at Naval Research Lab., Washington, D.C.
- (3) S.W. Golomb "Digital Communications with Space Applications". Prentice-Hall, (1964).
- (4) H.E. Chrestenson "A Class of Generalized Walsh Functions". Pacific Jour. of Math. Vol.5, pp. 17-31 (1955).
- (5) A.T. Butson "Generalized Hadamard Matrices" Proc. of Am. Math Soc. Vol.13, pp. 894-898 (1962).
- (6) J.J. Stiffler "Theory of Synchronous Communications". Prentice-Hall (1971).
- (7) S.H. Chang "A Note on Non-binary Orthogonal Codes" Digest of Technical Papers, IEEE International Communications Conference (1966) p. 278.
- (8) H.F. Harmuth "A Generalized Concept of Frequency and some Applications" IEEE Transactions on Information Theory, Vol.17-14, pp. 375-382, (1968).
- (9) D.A. Swick "Walsh Function Generation" IEEE Transactions on Information Theory, Vol. 11 15, No.1, January 1969, p. 167.
- (10) J.A. Chang "Ternary Sequency with Zero Correlations" Proc. of the IEEE July 1967, pp. 1211-1213.
- (11) B.L. van der Waerden "Modern Algebra" Vol.1 Translated by F. Blum and T.J. Secac, Frederick Ungar Publishing Co., New York (1950).
- (12) N.A. Alexandris "Relations Among Sequency, Axis Symmetry, Period of Walsh Functions" IEEE Information Theory, Vol.17, pp. 495-497 (1971).

POSITIVE REAL FUNCTIONS FROM ORTHOGONAL FUNCTIONS

N. K. Bose
Department of Electrical Engineering
University of Pittsburgh
Pittsburgh, Pa. 15213

Introduction

The properties and applications of orthogonal functions like Bessel functions, trigonometric functions, spherical functions, orthogonal polynomials,² Haar functions and Walsh functions³ in various engineering applications is well known. Besides the popular trigonometric functions, more general classes of complete systems of orthogonal functions have been used for theoretical investigations as well as equipment design, in recent years. Particular areas in which complete sets of orthogonal functions have been applied are as diverse as coding and multiplexing,^{3,10} signal processing,^{4,24} and general applications,⁵ vocoding, filtering, radiation, pattern recognition and image processing, and spectroscopy. Different sets of orthogonal functions have some underlying common properties. For example, most of the generally used systems of orthogonal functions are defined by linear differential or difference equations of second degree.

In 1967, Reza and Bose,¹¹ made use of an interesting mathematical property—interlacing of zeros of successive polynomials in a set of orthogonal polynomials like Hermite's, Legendre's, Laguerre's, etc.—to generate two element kind positive real functions. A. M. Soliman¹² extended the foregoing idea to some other orthogonal functions. In view of the overriding influence of orthogonal functions in various branches of science, it is deemed fit to make available the results to date on this aspect of the subject to the interested readers.

Proofs For Some Generated P. R. F.'s

The proofs for the generated positive real functions can be given by using the principle of mathematical induction or otherwise. Theorem 1 is proved using the induction principle.

Theorem 1. The real rational function, $\frac{jH_n(js)}{H_{n+1}(js)}$, where $H_n(s)$ is the Hermite polynomial of n th order, is a LC driving point function.

Proof: The recurrence relation¹³ valid

for $H_n(s)$ is (for $n > 0$)

$$H_{n+2}(s) - 2s H_{n+1}(s) + (2n+2)H_n(s) = 0,$$

with $H_0(s) = 1$, and $H_1(s) = 2s$.

Evidently, $\frac{jH_0(js)}{H_1(js)} = \frac{1}{2s}$ is a LC driving

point function. From the recurrence formula,

$$\frac{H_{n+2}(js)}{jH_{n+1}(js)} = 2s + \frac{jH_n(js)}{H_{n+1}(js)}(2n+2)$$

Thus, if $\frac{jH_n(js)}{H_{n+1}(js)}$ is LC realizable, so

also are $\frac{H_{n+2}(js)}{jH_{n+1}(js)}$ and its reciprocal,¹⁴

$$\frac{jH_{n+1}(js)}{H_{n+2}(js)}$$

Though the principle of mathematical induction can be used with facility, an alternate approach will be adopted to prove Theorem 2,

Theorem 2. The real rational function, $Z_{LC}(s) = \frac{jP_n(js)}{P_{n+1}(js)}$ is a driving-point reactance function, where $P_n(s)$ is the Legendre polynomial of n th order.

Proof: The Legendre polynomials, $P_n(s)$, $n = 0, 1, 2, \dots$ as defined¹⁵ by Rodrigues' formula, are

$$P_n(s) = \frac{1}{2^n n!} \frac{d^n (s^2 - 1)^n}{ds^n}$$

It is readily seen that $P_n(s)$ has n real roots, all lying between -1 and $+1$. Consequently, $P_n(js) = 0$ has n imaginary roots, all lying between $-j$ and $+j$. For odd values of n ,

$$-jP_n(js) = \frac{1}{2^n n!} \frac{d^n [(s^2 + 1)^n]}{ds^n}$$

$$-P_{n+1}(js) = \frac{1}{2^{n+1} (n+1)!} \frac{d^{n+1} [(s^2 + 1)^{n+1}]}{ds^{n+1}}$$

$$= \frac{1}{2^n n!} \frac{d^n [s(s^2 + 1)^n]}{ds^n}$$

whence $-[P_{n+1}(js)] + [-j P_n(js)] =$

$$\frac{1}{2^n n!} \frac{d^n [(s+1)(s^2+1)^n]}{ds^n} \text{ clearly, } -P_{n+1}(js)$$

and $-j P_n(js)$, are, for odd n , the even and odd parts of a real Hurwitz polynomial. An analogous procedure is adopted for even n to arrive at a similar result. This implies that $j P_n(js)$ is a reactance function. $\frac{j P_n(js)}{P_{n+1}(js)}$

Orthogonal Functions and Network Types

The table gives the generated positive real functions (p.r.f.'s) from corresponding orthogonal functions, and includes for completion, those reported earlier by Reza and Bose. It is seen that a wide variety of networks are characterized by the generated p.r.f.'s.

In the table, the first eleven driving-point functions characterize lumped networks, and the remaining characterize, in general, lumped-distributed networks. In particular, the first seven sets of generated functions characterize LC networks, the eighth characterizes a RLC network, and the ninth, tenth and eleventh correspond to the driving point impedances of RC networks. It is interesting to note that, though, the Tschebyscheff polynomials, Gegenbauer polynomials, and the Legendre polynomials, which are all special cases of the hypergeometric polynomials of Jacobi, can be used to generate p.r.f.'s in the manner just indicated, the Jacobi polynomials, in general, cannot be so used. This follows from the fact that the location of the roots of the Jacobi polynomials are not symmetric with respect to the origin. Interestingly enough, the twelfth generated function is a positive real function of several variables, which are being used more and more in the synthesis of lumped-distributed networks.^{19,20} The last three generated functions, obviously characterize distributed networks. The results concerning the Walsh functions can be obtained using the difference equations defining them.

It must be borne in mind, that other p.r.f.'s can be generated from the given orthogonal functions, with equal facility. The procedures for generation, among those available in the literature, will depend upon the desired types of applications to which the generated functions are to be put. It appears that many useful properties of orthogonal functions can be quickly proved by constructing suitable network models using the generated functions, as suggested by Reza and Bose, earlier, and also more recently discussed by a couple of authors.^{22,23}

Table

Name of Orthogonal Function	Generated P.R.F.'s
1. Legendre polynomials, $P_n(s)$.	$j \frac{P_n(js)}{P_{n+1}(js)}$
2. Associated Legendre spherical harmonic of the first kind, $P_n^m(s)$.	$j \frac{P_n^m(js)}{P_{n+1}^m(js)}$
3. Gegenbauer or modified ultraspherical polynomials, $C_n^m(s)$.	$j \frac{C_{n+1}^m(js)}{C_{n+1}^m(js)}$
4. Tschebyscheff polynomial of the first kind, $T_n^{(1)}(s)$.	$j \frac{T_n^{(1)}(js)}{T_{n+1}^{(1)}(js)}$
5. Tschebyscheff polynomial of the second kind, $T_n^{(2)}(s)$.	$j \frac{T_n^{(2)}(js)}{T_{n+1}^{(2)}(js)}$
6. Hermite polynomial $H_n(s)$.	$j \frac{H_n(js)}{H_{n+1}(js)}$
7. Parabolic Cylinder function, $D_n(s)$.	$j \frac{D_n(js)}{D_{n+1}(js)}$
8. Bessel polynomial, $B_n(s)$.	$s \frac{B_{n-1}(s)}{B(s)}$
9. Laguerre polynomial, $L_n(s)$.	$\frac{L_n(-s)}{L_{n+1}(-s)}$
10. Uspensky ¹¹ polynomial, $U_n(s)$.	$\frac{U_n(s)}{U_{n+1}(s)}$
11. Second-order Q-polynomial, ¹⁶ $Q_n^{(2)}(s)$.	$\frac{Q_n^{(2)}(s)}{Q_{n+1}^{(2)}(s)}$
12. n-variate Hermite ¹⁷ polynomial, $H_{1234}^{(n)}(x_1, x_2, x_3, \dots, x_n)$.	$j \frac{H_{1--n}^{(n)}(-jx_1, \dots, -jx_n)}{H_{1--n+1}^{(n+1)}(-jx_1, \dots, -jx_{n+1})}$
13. Bessel's function $J_n(s)$.	$j \frac{J_{n-1}(js)}{J_n(js)}$
14. Fourier transforms of Walsh ¹⁸ pulses $jS(n, f)$, $C(n, f)$.	$-\frac{C(n, jf)}{S(n+1, jf)}$

15.

$$-\frac{C(n+1, jf)}{S(n+1, jf)}$$

Conclusions

It has been shown, that using the principle of mathematical induction or otherwise, it is possible to generate positive real functions from successive functions in sets of orthogonal functions. The procedure for generation emphasized here depends upon the interlacing property of zeros of the relevant functions. Other methods of generation are also possible using classical techniques, depending upon the nature of application of the generated functions. In view of the very widespread present day use of orthogonal functions like Haar, Rademacher, and Walsh functions,^{25,26} it might be useful to study these functions using network models obtained from their generated functions.

References

1. P. M. Morse and H. Feshbach, "Methods of theoretical physics," McGraw-Hill Book Co., Inc., New York, 1953, vol. 1, pp. 781-787; vol. 2, pp. 1754-1756.
2. H. F. Harmuth, "Application of Walsh functions in communication," IEEE Spectrum, Nov. 1969, pp. 82-89.
3. H. D. Lueke, "Binäre Orthogonale signalalphabet mit speziellen Korrelationseigenschaften," Archiv. Elek. Übertragung, vol. 20, pp. 310-316, 1966.
4. J. L. Hammond and R. S. Johnson, "A review of orthogonal square-wave functions and their application to linear networks," Journal of Franklin Inst., 1962, pp. 221-225.
5. G. F. Sandy, "Square-wave (Rademacher-Walsh functions) analysis," Report #WP-1585, Mitre Corporation, 1820 Dolley Madison Blvd., McLean, Virginia, October 1968.
6. H. F. Harmuth, "Sequence filters based on Walsh functions," IEEE Trans. on Electromagnetic Compatibility, vol. 10, 1968, pp. 293-295.
7. H. F. Harmuth, "Transmission of information by orthogonal functions," Springer-Verlag, Berlin/ New York, 1969.
8. B. Meltzer, "Speculations on the use of orthogonal functions in the study of morphogenesis," Nature, Vol. 217, January 1968, p. 13.
9. J. A. Decker and M. Harwit, "Experimental operation of a Hadamard spectrometer," Applied Optics, vol. 8, no. 12, December 1969, pp. 2552-2554.
10. H. F. Harmuth, "A generalized concept of frequency and some applications," IEEE Trans. on Information Theory, vol. 14, May 1968, pp. 375-382.
11. F. M. Reza and N. K. Bose, "Some links between theory of equations and realizability theory," Proceedings of Tenth Midwest Symposium on Circuit Theory, May 1967, pp. v-2-1 to v-2-11.
12. A. M. Soliman, "Theory of multi-variable positive in distributed network synthesis," Ph.D. Dissertation, University of Pittsburgh, June 1970, pp. 30-33.
13. W. Magnus and F. Oberhettinger, "Formulas and theorems for the functions of mathematical physics," Chelsea Publishing Co., New York 1949, p. 83.
14. L. Weinberg, "Network analysis and synthesis," McGraw-Hill Book Co., Inc., New York, 1962, p. 244.
15. E. T. Whittaker and G. N. Watson, "A course of modern analysis," University Press, Cambridge, 1958, p. 303.
16. R. Deitch, "Nonlinear transformation of random processes," Prentice-Hall, Englewood Cliffs, N.J., 1962, pp. 80-82.
17. H. Grad, "Note on N-dimensional Hermite Polynomials," Communications on Pure and Applied Mathematics, 2 Dec. 1949, pp. 325-330.
18. H. H. Schreiber, "Bandwidth requirement for Walsh functions," Proceedings on Applications of Walsh Functions, 1970, pp. 46-51.
19. J. D. Rhodes and P. C. Marston, "Cascade synthesis of transmission lines and lossless lumped networks," Electronics Letters, vol. 7, no. 20, Oct. 1971, pp. 621-622.
20. N. K. Bose and A. M. Soliman, "A novel approach to the synthesis of multivariable positive real functions," Electronics Letters, vol. 5, no. 26, Dec. 1969, pp. 717-718.
21. N. K. Bose, "Class invariant generation of positive real functions," Proc. Allerton Conf. Circuit and System Theory, October 1968, pp. 256-260.

22. R. E. Vowels, "A note on positive real functions," IEEE Trans. on Circuit Theory, vol. CT-18, May 1971, pp. 383-386.

23. N. K. Bose, "A note on multi-variable positive real functions," to be published in IEEE Trans. on Circuit Theory, March 1972.

24. S. S. Penbeci and S. C. Lee, "Synthesis of distributed RC networks by means of orthogonal functions," IEEE Trans. on Circuit Theory, vol. CT-16, February 1969, pp. 137-140.

25. C. A. Bass, ed., "Application of Walsh functions," Proceedings of Symposium and Workshop, Naval Research Laboratory, Washington, D.C. 1970.

26. R. B. Schulz, ed., "Application of Walsh functions," IEEE Trans. on Electromagnetic Compatibility, vol. 13, August 1971.

27. R. J. Duffin, "Orthogonal sawtooth functions," Technical report published by the Carnegie Institute of Technology, ASTIA Document no. 252483, December 29, 1960.

INEQUIVALENT SETS OF 2^n TWO LEVEL ORTHOGONAL FUNCTIONS

Gustavus J. Simmons
Sandia Laboratories
Albuquerque, New Mexico U.S.A.

Introduction

The n^{th} order Kronecker product

$$R_n = \begin{pmatrix} u & v \\ v & u \end{pmatrix}^n = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}^n$$

is a $2^n \times 2^n$ matrix whose rows (columns) are the Rademacher-Walsh functions. Much of the utility of these functions lies in the fact that they are a complete orthonormal set--properties which are preserved in the equivalence class determined by the operations of:

1. multiplying each row (column) by the same binary code
2. permuting rows (columns)

i.e., $A \approx R_n$ if and only if there exist permutation matrices P and Q , whose nonzero entries may be either ± 1 , such that $A = PR_nQ$. One might suspect that all $2^n \times 2^n$ orthogonal arrays would be equivalent since there is only one equivalence class for $n = 1, 2$ or 3 [1,2]. However, this is not the case as is illustrated by the following 16×16 array which differs from R_n only in the last four rows but which cannot be transformed into R_n by any sequence of the operations (1) and (2).

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1
1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1
1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1
1	1	1	1	-1	-1	-1	-1	1	1	1	1	-1	-1	-1	-1
1	-1	1	-1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1
1	1	-1	-1	-1	1	-1	-1	1	1	-1	-1	1	1	-1	-1
1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1
1	1	1	1	1	-1	-1	-1	-1	1	1	1	1	-1	-1	-1
1	-1	1	-1	1	1	-1	-1	-1	1	-1	1	-1	1	-1	1
1	1	-1	-1	-1	1	-1	-1	1	1	-1	-1	1	1	-1	-1
1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1

Since the invariants of two dimensional patterns under plane transformations are dependent on the coordinate system in which the pattern is represented, the determination of inequivalent bases of 2^n two level functions is important in many applications, especially in pattern recognition studies. This paper develops a theoretical basis for constructing these bases.

The indexing system used in this paper to relate a linear vector to a two-dimensional pattern was selected so that truncation of the vector would result in a lower order pattern. The leftmost symbol of the vector is mapped into the upper left corner of the pattern, the next

symbol into the square to its right, etc., until the 2×2 upper left square is filled. The 2×2 square to the right is then filled in the same manner, etc., until the pattern is complete; i.e.,

$$X = x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10} x_{11} x_{12} x_{13} x_{14} x_{15} x_{16}$$

becomes

$$\begin{matrix} x_1 & x_2 & x_5 & x_6 \\ x_3 & x_4 & x_7 & x_8 \\ x_9 & x_{10} & x_{13} & x_{14} \\ x_{11} & x_{12} & x_{15} & x_{16} \end{matrix}$$

Group properties of R_n

Many of the combinatorial properties of the arrays R_n are a consequence of the fact that the rows are also elements of a multiplicative involutory group of order 2^n .

Since there is no possibility of confusion, we shall represent both the array and the group by R_n . The group multiplication is defined by an element by element composition in $GF(2)$ which can be replaced by an equivalent operation on the literals u and v in those cases in which the product is between rows of R_n which are designated by products of u and v . For example, $(u^3v) \cdot (uvuv) = uvu^2$, etc. u^n is the identity for R_n . Obviously, every element, $r_i \in R_n$, is its own inverse

$$r_i \cdot r_i = u^n \quad (1)$$

i.e., all elements except the identity are of order two, which is the definition of an involutory group. All involutory groups are necessarily abelian.

Since every element r_i of R_n ($r_i \neq u^n$) is of order two;

$$\langle i = (u^n, r_i) \quad r_i \in R_n$$

is a subgroup of R_n of order two. There are $2^n - 1$ such subgroups. If G_1 is extended by any other element r_j , $r_j \neq r_i$, the new subgroup is of order 4, and the number of subgroups of order 4 is

$$\#(G_1) = \frac{(2^n - 1)(2^{n-1} - 1)}{1 \cdot 3}; \quad |G_1| = 4 \quad (2)$$

This method of generating subgroups can be generalized to all orders 2^a , $a \leq n$. If a independent generators are to be selected, the first can be selected in $2^n - 1$ ways, the second in 2^{n-2} ways, etc. But the ordering of the a generators can be effected in $(2^a - 1)!$ ways for the first, $(2^{a-2} - 2)!$ for the second, etc. The number of subgroups of order 2^a in R_n is

therefore given by the well-known formula [3]:

$$\#(G_1) = \frac{(2^{\alpha}-1)(2^{\alpha}-2)(2^{\alpha}-4)\dots(2^{\alpha}-2^{\alpha-1})}{(2^{\alpha}-1)(2^{\alpha}-2)(2^{\alpha}-4)\dots(2^{\alpha}-2^{\alpha-1})} \quad (3)$$

where

$$|G_1| = 2^{\alpha}$$

The vectors with norm 1, i.e., points on the surface of the unit hypersphere, we call figures. If they are also a vertex of the unit hypercube centered at the origin with edges parallel to the coordinate axes we call the figure a pattern, i.e., a pattern is one of the 2^{α} binary vectors.

All of the subgroups of R_n are necessarily direct products of subgroups of order two, i.e., if $r_i \neq r_j$

$$(1, r_i) \times (1, r_j) = (1, r_i, r_j, r_i r_j)$$

since R_n is an involutory group. We define a vector function P_S on the elements of a set S of rows from R_n :

$$P_S = u^n - 2^{1-N} \prod_{r_i \in S} (u^n - r_i) \quad (4)$$

where

$$|S| = N$$

In particular Equation (4) reduces to the following for the j^{th} element of the vector;

$$P_S(j) = 1 - 2^{1-N} \prod_{r_i \in S} (1 - r_i(j)) \quad (5)$$

Theorem 1.

P_S , as defined by Equation (4), is a pattern.

Proof: If any r_i has a +1 entry at the j^{th} position, the corresponding factor $(1 - r_i(j))$ in Equation (5) is zero and consequently the j^{th} entry in P_S is a +1. If all r_i entries in the j^{th} position are -1, the product in Equation (5) is 2^N , and the j^{th} entry in P_S is a -1. But any vector with only ± 1 entries is a pattern. ■

Corollary 1.

Obviously, P_S has a -1 entry if and only if all of the corresponding positions in the r_i are -1.

Corollary 2.

If $|S| = 1$, $P_S = r_1$ where $S = \{r_1\}$.

Corollary 2 shows that the set S must contain more than one element if P_S is to be a pattern not already in S .

* Professor D. R. Morrison of the University of New Mexico has suggested that this be called a generalized circle function because of its similarity to the classical relation among the n^{th} roots of unity.

Theorem 2.

The pattern P_S is orthogonal to $\{R_n - [S]\}$ and is not orthogonal to any element in $[S]$, for $|S| > 1$.

Proof: From Equation (4) we see that P_S can be expressed as a linear combination (uniquely since R_n is complete) of terms in R_n in the form:

$$P_S = \frac{2^{N-1}-1}{2^{N-1}} u^n + \frac{1}{2^{N-1}} \left\{ \sum_{i \in S} r_i - \sum_{\substack{i \in S \\ j \in S \\ i \neq j}} r_i r_j + \dots + (-1)^{N-1} r_1 r_2 \dots r_N \right\} \quad (6)$$

The proof of Theorem 2 follows immediately from Equation (6) by noting that

$$(P_S, u^n) = \frac{2^{N-1}-1}{2^{N-1}} \neq 0 \quad \text{for } N > 1$$

Every element in $[S]$ is a product of elements in S , however every such product appears in the bracketed expression of Equation (6). Remembering that the rows r_i are elements of R_n , and as such are orthogonal by pairs we get finally:

$$r \in [S], \quad r \neq u^n$$

imply

$$(P_S, r) = \frac{1}{2^{N-1}} \neq 0 \quad \text{for any } N.$$

Similarly, if $r \in \{R_n - [S]\}$, then $(P_S, r) = 0$ since all of the inner products under the summation signs in Equation (6) are identically 0. ■

P_S partitions R_n into those elements orthogonal to P_S and those which are not; i.e., the subgroup $[S]$ generated by S . We wish to strengthen this construction.

Lemma: If the subset S contains any two elements r_i and r_j and their product $r_i r_j$, then P_S is identically u^n .

Proof: By Corollary 1 to Theorem 1, P_S has a -1 entry if and only if all of the factors in Equation (5) are -1, however, it is impossible for r_i , r_j and $r_i r_j$ to be simultaneously -1. Therefore P_S is +1 in all positions if the product of any two elements in S is also in S . ■

If $P_S = u^n$, the construction given above does not lead to any elements not in S already-- which was the reason for introducing P_S in the first place. We can complete the development by placing appropriate constraints on the subset S . Since $[S]$ is a subgroup of R_n , we know that $|[S]| = 2^{\alpha}$ where $\alpha \geq 1$ when $|S| > 1$. Furthermore, since R_n is an involutory group it is possible to select α generators for $[S]$ from $[S]$. In fact there are Q sets of generators

$$Q = (2^{\alpha}-1)(2^{\alpha}-2)(2^{\alpha}-2^2)\dots(2^{\alpha}-2^{\alpha-1})$$

for S.

Given any subgroup $G \subset R_n$, we define a generator set S' with $\log_2 |G|$ elements in it. P_G is the vector function P_g given by Equation (4). P_G has the following properties, where $|G| \geq 4$:

1. P_G is a pattern Theorem 1
2. If $g \in G$,
 $(P_G, g) \neq 0$ Theorem 2
3. If $h \in \{R_n - G\}$
 $(P_G, h) = 0$ Theorem 2
4. $P_G \notin R_n$

Point (4) follows since $P_G \notin \{R_n - G\}$, and

$$(P_G, u^n) = \frac{2^{N-1}-1}{2^{N-1}} \Rightarrow P_G \neq u^n$$

and if $g \in G$, $g \neq u^n$

$$(P_G, g) = \frac{1}{2^{N-1}} \Rightarrow P_G \neq g$$

i.e., $P_G \notin G$.

Theorem 3.

The inner product commutes with the vector product if there is a common factor:

$$(xy, zy) = (x, z)(y, y) = (x, z)$$

Theorem 3 is useful primarily because it leads to the following corollary concerning P_G .

Corollary 1.

If G is a subgroup of R_n , $|G| \geq 4$, and if $g, h \in G$, $g \neq h$, then;

$$(gP_G, hP_G) = 0$$

and further if $g \in G$, and $h \in \{R_n - G\}$

$$(gP_G, h) = 0$$

Proof: The first part of the corollary follows by letting $y = P_G$ in Theorem 3. The second part is a simple consequence of Theorem 2 and the fact that G is a group. Theorem 2 showed (in its proof) that P_G could be expressed as a linear expansion in terms of the elements in G , and hence

$$(P_G, h) = 0$$

if $h \in \{R_n - G\}$. But gP_G is still a linear expression in terms of the elements of G , and therefore

$$(gP_G, h) = 0$$

if $g \in G$ and $h \in \{R_n - G\}$. ■

The preceding corollary is fundamental to much of the following development, so we shall digress for a moment to illustrate the construction. Let $G = \{u^4, vu^2v, u^2vu, vuv^2\}$



Subgroup G

and let $S = \{vu^2v \text{ and } u^2vu\}$, then P_G is the vector

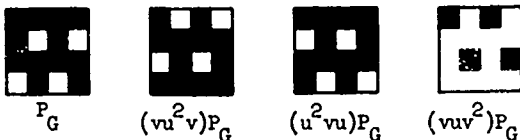
1 1 1 -1 1 1 1 -1 1 1 -1 1 1 1 -1 1
or the pattern:



with the spectrum (in I_n):

$$P_G = \frac{1}{2}(1 \ 0 \ 1 \ 0, \ 0 \ 0 \ 0 \ 0, \ 0 \ 1 \ 0 \ -1, \ 0 \ 0 \ 0 \ 0)$$

The coset gP_G ; $g \in G$ is shown by the following figure:



gP_G

It is easily verified that these patterns are mutually orthogonal.

If

$$|G| = 2^a$$

the number of cosets of G is 2^{n-a} . The cosets of G possess a very useful property, summarized in the following theorem.

Theorem 4.

Let $x \in \{R_n - G\}$, and Gx be the coset formed by x on G . Similarly, GP_G is the coset formed by P_G . The set of patterns xGP_G have the following properties

1. If $g, h \in xGP_G$ $g \neq h$
then $(g, h) = 0$
2. If $g \in xGP_G$ and $h \in \{R_n - Gx\}$
then $(g, h) = 0$

Proof: First, it is obvious that xGP_G consists of patterns, and not figures, since the product of patterns is a pattern. If g and h are elements of xGP_G , they must be of the form

$$\left. \begin{aligned} g &= xg_1P_G \\ h &= xg_2P_G \end{aligned} \right\} g_1, g_2 \in G$$

and therefore

$$\begin{aligned}
(g, h) &= (xg_1 P_G, xg_2 P_G) \\
&= (g_1 x P_G, g_2 x P_G) \\
&= (g_1, g_2) \quad \text{by Theorem 3} \\
&= 0
\end{aligned}$$

since $G \subset R_n$ and $g_1 \neq g_2$. The second part of the proof is somewhat more difficult. Again since $g \in xGP_G$, we have

$$g = xg_1 P_G; \quad g_1 \in G$$

and since $h \in \{R_n - Gx\}$

$$h = r_1$$

There are two cases to be considered; $h \in G$ (which is possible since $G \cap Gx = \emptyset$ when $x \in \{R_n - G\}$) and $h \in yG$ where $y \in \{R_n - (G \cup Gx)\}$. If

$$G \cup Gx = R_n$$

i.e., if $|G| = 2^{n-1}$, then there cannot be an h of the second type.

Case I $h \in G$

$$(h, xg_1 P_G) = (g_1 g_2, xg_1 P_G)$$

where $g_1, g_2 \in G$, since G is a subgroup. By Theorem 3

$$(h, xg_1 P_G) = (g_2, xP_G) = (xxg_2, xP_G)$$

using the fact that the group is involutory. Therefore,

$$(h, xg_1 P_G) = (xg_2, P_G) = (r_1, P_G)$$

where $r_1 = xg_2 \in Gx$, i.e., $r_1 \in \{R_n - G\}$. But we have already proven in the corollary to Theorem 3 that this last inner product is 0; therefore,

$$(h, xg_1 P_G) = 0$$

Case II $h \in \{R_n - (G \cup Gx)\}$

$$(h, xg_1 P_G) = (yg_1, xg_1 P_G)$$

where $g_1 \in G$ and $y \in \{R_n - (G \cup Gx)\}$

$$(h, xg_1 P_G) = (xh, g_1 P_G)$$

We must show that $xh \notin G$, i.e., that $xh \in \{R_n - G\}$. But we know by construction that

$$h \in \{R_n - (G \cup Gx)\}$$

therefore

$$xh \in \{R_n - (G \cup Gx)\} \subseteq \{R_n - G\}$$

since $x = r_1 \in R_n$. And finally by use of the corollary to Theorem 3 again we find

$$(h, xg_1 P_G) = 0. \blacksquare$$

Theorem 4 is the essential result needed to construct inequivalent bases for the pattern space. The theorem says that we can take any subgroup G , of R_n , from the set of patterns GP_G and then replace the coset Gx of R_n with a new

set of orthonormal pattern vectors xGP_G and still have a complete orthonormal basis. This result is true in general since no restriction was made on which coset Gx was to be selected except to insist that it be a proper coset-- $Gx \neq G$. This guarantees that the procedure just described will generate new bases, however, these may be equivalent so we require finer techniques to deal with the problem of generating inequivalent bases.

We digress from the main development for a moment to give some results which help explain the construction of Theorem 4. We noted earlier in connection with the development for Theorem 3 that P_G has $2^{n-\alpha}$ -1's in it where $\alpha = |S|$ and S was any one of the generating sets for G . We now note a surprising result.

Corollary

In spite of the fact that P_G has only $2^{n-\alpha}$ -1's, all of the replacement vectors (for Gx) in xGP_G have 2^{n-1} +1's and 2^{n-1} -1's.

Proof: We could prove this directly, however, we have already proven in Theorem 4 that any element of xGP_G is orthogonal to any element of G , and hence to u^n in particular. But u^n consists of 2^n +1's, and any vector orthogonal to it consists of half +1's and half -1's. ■

The example which we used earlier was

$$G = \{u^4, vu^2v, u^2vu, vuv^2\}$$

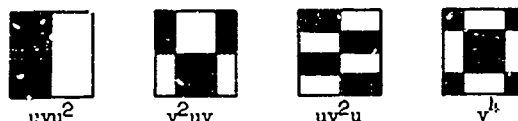
which has three proper cosets

$$C_1 = \{uvu^2, v^2uv, uv^2u, v^4\}$$

$$C_2 = \{vu^3, u^3v, vuvu, u^2v^2\}$$

$$C_3 = \{v^2u^2, uvuv, v^3u, uv^3\}$$

Consider coset C_1 and the associated set of replacement vectors, say $(uvu^2)GP_G$;



C_1



$\{uvu^2\}GP_G$

$(uvu^2)GP_G$ exhibits all of the properties which we have derived for the replacement sets. However, we selected C_1 as our example because it contains uvu^2 which is the third row in the block normalized R_n array. We wish to determine whether the new array obtained by replacing C_1 by $(uvu^2)GP_G$ can also be put in block normal form. The following array exhibits r_1 and r_2 , r_4 and r_5 and $(uvu^2)GP_G$ to illustrate the procedure for constructing the permutation which will return the array to block normal form.

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1
1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1
1	1	1	-1	-1	-1	1	1	-1	-1	-1	1	1	-1	-1	1
1	-1	1	1	-1	1	-1	1	1	1	1	1	-1	-1	-1	1
1	1	-1	1	-1	-1	1	-1	1	1	1	-1	-1	-1	1	1
1	-1	-1	-1	1	1	1	1	-1	-1	-1	1	-1	1	1	1

Since xGP_G is to be substituted for xG , $(uvu^2)P_G$ is in the new array. The other rows of xGP_G could be used equally well however the method is valid in general irrespective of which coset is replaced. If columns 4 and 8 are interchanged and columns 11 and 15 the rows r_1, r_2, r_4 and r_5 will be unaffected and the row labeled uvu^2P_G will have been transformed to be uvu^2 , i.e., by bringing the permuted row v^4 to row three we have returned the array to block normal form. In the next theorem we shall prove that this was not a coincidental property of the construction, but rather a consequence of including a row from the first $n+1$ rows of the block normal form in G .

In the following discussion we shall frequently wish to refer to the block of $n+1$ rows at the top of a block normalized array. Since these rows have a fixed form--independent of the structure of the array in which they occur--we can properly refer to the block B_n as the first $n+1$ rows in the block normal ordering. Also since most of the remaining discussion is concerned with the orthonormal set of vectors generated by replacing the coset Gx in R_n with the set of vectors xGP_G , we need a symbol for this array. Let

$$Q(Gx) = R_n \cup xGP_G - Gx$$

which is a complete specification of the array since both xGP_G and Gx depend only on the subset G and on some element x from the coset.

The main theorem which restricts the cosets so that the Q are inequivalent bases can now be given.

Theorem 5.

If the coset Gx which is replaced with the vector set xGP_G to form $Q(Gx)$ includes an element $b \in B_n$, then $Q(Gx)$ can be put in block normal form by an appropriate permutation of its rows and columns.

Proof:

Case I

If G does not contain a term involving b , then P_G is a linear combination of terms not involving b and which can therefore be permuted in any way which does not affect $\{B-b\}$. One of these permutations represents each term in bGP_G , and therefore we can return $Q(Gx)$ to block normal form.*

* As was shown in the preceding example any of the elements $(uvu^2)P_G$, $(v^2uv)P_G$, $(uv^2u)P_G$ or $(v^4)P_G$ could have been permuted to return $Q(Gx)$ to block normal form.

Case II

If G contains a term (or terms) which involve b as a factor, we must use a different argument. First, we illustrate this case with an example. As before let

$$G = (b_1, b_2 b_5, b_4, b_2 b_4 b_5)$$

however, form the coset Gb_2

$$Gb_2 = (b_2, b_5, b_2 b_4, b_4 b_5)$$

In this case both b_2 and b_5 are removed from the array in forming $Q(Gx)$ since both are in the coset Gb_2 . Therefore P_G is not a linear combination of terms of B remaining in $Q(Gx)$ and no permutation is possible on bGP_G to return $Q(Gx)$ to block normal form. Instead, we use the following procedure. If $|G| < 2^{n-1}$, then G and Gb did not exhaust R_n and there is at least one $b_1 \in B$ which does not appear in either G or Gb_2 . The products $b_1 b_{j_1}, b_1 b_{j_1} b_{j_2}, \dots$ where the b_j are the elements of B which appear in Gb_2 are therefore in the set $\{R_n - G \cup Gb_2\} \subseteq Q(Gx)$. But these terms can be simultaneously permuted to recover the b_1 without affecting the rows of $B \cap Q(Gx)$ in the same manner as was done with P_G in Case I. ■

In the example given in the discussion of Case II in proving Theorem 5 we had

$$B \cap Q(Gx) = b_1, b_3, b_4$$

with b_2 and b_5 as elements in the coset. We derive the normalizing permutation for this case to illustrate the method. $b_2 b_3 = uv^2$ and $b_2 b_3 b_5 = uvu^2$ are both in $Q(Gx)$. The first three rows of the following array are b_1, b_3 and b_4 and the last two are $b_2 b_3$ and $b_2 b_3 b_5$.

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1
1	1	1	1	-1	-1	-1	-1	1	1	1	1	-1	-1	-1	-1
1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1
1	-1	-1	1	1	-1	-1	1	-1	1	-1	-1	-1	1	1	-1

The column permutation to convert $b_2 b_3 b_5$ to b_5 and $b_2 b_3$ to b_2 simultaneously, while leaving b_1, b_3 and b_4 unchanged is

$$(1)(2,10,9)(3,12,4)(5)(6,14,13)(11)(7,16,8)(15).$$

This permutation is completely determined by the choice of elements in $Q(Gb)$ which are to be reordered. We could have equally well selected $b_3 b_5$ or $b_2 b_3 b_5$, but we could not select $b_2 b_4$ since it was an element of Gb_2 and therefore not in $Q(Gb)$.

Theorem 5 defines the problem: to determine those arrays $Q(Gx)$ generated by the replacement of cosets for which

$$B \cap Gx = \emptyset. \quad (7)$$

Cosets for which

$$B \cap Gx \neq \emptyset$$

we shall call denied, in the sense that the arrays which they generate can be reduced to some other block normal form, i.e. they cannot be inequivalent block normal arrays.

Subgroup structure of R_n

Theorem 5 of the preceding section showed that only cosets Gx for which

$$B \cap Gx = \emptyset \quad (8)$$

need be considered in the search for inequivalent bases. In this section we shall develop necessary and sufficient conditions for Equation (8) to be satisfied, and enumerate the resulting bases for E_{16} .

Theorem 6.

The only bases for E_2 , E_4 , E_8 and E_{16} which are also groups are in the equivalence classes of R_1 , R_2 , R_3 and R_4 , respectively.

Proof: The conclusion is trivial for E_2 , E_4 and E_8 since the only bases in those cases are equivalent to R_1 , R_2 and R_3 . In the case of E_{16} , we have proven that all bases are equivalent to some normal array, i.e., an array which includes B . However

$$[B] = R_n$$

and the theorem follows. ■

Corollary 1.

If the subgroup G of R_n is of order 2^{n-1} , its cosets cannot satisfy Equation (8), i.e., they cannot be disjoint from B .

Proof: $B \not\subset G$, for if it were G would be R_n by Theorem 6. Therefore the only coset, $Gx \neq G$, has a non-empty intersection with B

$$Gx \cap B \neq \emptyset. \blacksquare$$

Corollary 2.

E_{16} is the smallest space which could have an equivalent pair of bases.

Proof: From Corollary 2 to Theorem 1, we know that G must be of order four or greater if the method of generating new bases described by Theorem 4 is to lead to a new base. But Corollary 1 says that the cosets of a subgroup of order 4 cannot have a member which is disjoint from B unless $n \geq 4$, i.e., for E_{16} or larger spaces. ■

The important point of the preceding argument is that we need only consider the subgroups of R_4 of order 4 in constructing inequivalent bases for E_{16} . Equation (3) gave the number of such subgroups to be 35. These are tabulated in Table 1.

The meaning of the subgroup types can best be illustrated by examples. Consider first, the subgroup G_1 and its cosets:

$$\begin{aligned} G_1 &= (u^4, u^3v, u^2vu, u^2v^2) \\ G_1(uvu^2) &= (uvu^2, uvuv, uv^2u, uv^3) \\ G_1(vu^3) &= (vu^3, vu^2v, vuuv, vuuv^2) \\ G_1(v^2u^2) &= (v^2u^2, v^2uv, v^3u, v^4) \end{aligned}$$

Subgroups of R_4 of order 4

Subgroup	Composition	Type
G_1	$(u^4, u^3v, u^2vu, u^2v^2)$	I
G_2	$(u^4, u^3v, uvu^2, uvuv)$	I
G_3	(u^4, u^3v, vu^3, vu^2v)	I
G_4	$(u^4, u^2vu, uvu^2, uv^2u)$	I
G_5	$(u^4, u^2vu, vu^3, vuuv)$	I
G_6	$(u^4, uvu^2, vu^3, v^2u^2)$	I
G_7	(u^4, u^3v, uv^2u, uv^3)	II
G_8	$(u^4, u^3v, vuuv, vuuv^2)$	II
G_9	$(u^4, u^3v, v^2u^2, v^2uv)$	II
G_{10}	$(u^4, u^2vu, uvuv, uv^3)$	II
G_{11}	$(u^4, u^2vu, vu^2v, vuuv^2)$	II
G_{12}	$(u^4, u^2vu, v^2u^2, v^3u)$	II
G_{13}	$(u^4, uvu^2, u^2v^2, uv^3)$	II
G_{14}	$(u^4, uvu^2, vu^2v, v^2uv)$	II
G_{15}	$(u^4, uvu^2, vuuv, v^3u)$	II
G_{16}	$(u^4, vu^3, u^2v^2, vuuv^2)$	II
G_{17}	$(u^4, vu^3, uvuv, v^2uv)$	II
G_{18}	(u^4, vu^3, uv^2u, v^3u)	II
G_{19}	(u^4, u^3v, v^3u, v^4)	III
G_{20}	(u^4, u^2vu, v^2uv, v^4)	III
G_{21}	$(u^4, uvu^2, vuuv^2, v^4)$	III
G_{22}	(u^4, vu^3, uv^3, v^4)	III
G_{23}	$(u^4, u^2v^2, uvuv, uv^2u)$	IV
G_{24}	$(u^4, u^2v^2, vu^2v, vuuv)$	IV
G_{25}	$(u^4, uvuv, vu^2v, v^2u^2)$	IV
G_{26}	$(u^4, uv^2u, vuuv, v^2u^2)$	IV
G_{27}	$(u^4, u^2v^2, v^2u^2, v^4)$	V
G_{28}	$(u^4, uvuv, vuuv, v^4)$	V
G_{29}	(u^4, uv^2u, vu^2v, v^4)	V
G_{30}	$(u^4, u^2v^2, v^2uv, v^3u)$	VII
G_{31}	$(u^4, uvuv, vuuv^2, v^3u)$	VII
G_{32}	$(u^4, uv^2u, vuuv^2, v^2uv)$	VII
G_{33}	(u^4, uv^3, vu^2v, v^3u)	VII
G_{34}	$(u^4, uv^3, vuuv, v^2uv)$	VII
G_{35}	$(u^4, uv^3, vuuv^2, v^2u^2)$	VII

Table 1

The intersections of these cosets with B are given by the following expressions, in which the coset leader for cosets 2 and 3, happens to also be the intersection.

$$\begin{aligned}
G_1 \cap B &= (u^4, u^3v, u^2vu) \\
G_1(uvu^2) \cap B &= (uvu^2) \\
G_1(vu^3) \cap B &= (vu^3) \\
G_1(v^2u^2) \cap B &= \emptyset
\end{aligned}$$

Therefore only the last coset can generate a non-equivalent base for E_{16} .

On the other hand consider the subgroup G_{19} and its cosets:

$$\begin{aligned}
G_{19} &= (u^4, u^3v, v^3u, v^4) \\
G_{19}(uvu^2) &= (uvu^2, uvuv, vuvu, vuv^2) \\
G_{19}(vu^3) &= (vu^3, vu^2v, uv^2u, uv^3) \\
G_{19}(v^2u^2) &= (v^2u^2, v^2uv, u^2vu, u^2v^2)
\end{aligned}$$

The intersections of these cosets with B are

$$\begin{aligned}
G_{19} \cap B &= (u^4, u^3v) \\
G_{19}(uvu^2) \cap B &= (uvu^2) \\
G_{19}(vu^3) \cap B &= (vu^3) \\
G_{19}(v^2u^2) \cap B &= (u^2vu)
\end{aligned}$$

hence G_{19} cannot be used to generate new non-equivalent bases for E_{16} . Thus there exist subgroups of order 4 which lead to new bases and others which do not.

The reason for the different behavior of G_1 and G_{19} is that their cosets span 12 and 16 elements of R_4 respectively. For example, if $uvuv$ is in G_1 , then any coset which includes u^3v , uvu^2 , $uvuv$, uv^3 or v^2uv will have a non-empty intersection with B and hence by Theorem 5 could not lead to a new base for E_{16} . In the case of G_{19} all elements other than v^2u^2 , v^2uv , v^3u and v^4 are denied as coset members; but these four are themselves a coset of G_1 which can be replaced by xG_{19} to form a new basis. On the other hand G_{19} spans all elements of R_4 and no such coset is possible.

Every element in G has $n+1$ factors which it denies as members of cosets, since the product would be in B . The elements u^4 and v^4 each deny 5 elements as terms in cosets, if the cosets are to be disjoint from B , but none of the five terms denied by u^4 are included in the set of five denied by v^4 . On the other hand, u^4 and u^3v together deny only eight terms. It is this overlapping of denied elements which makes it possible for the four terms in some subgroups to deny only twelve factors, as was the case for G_1 , and hence to leave a coset disjoint from B . On the other hand for a space of suitably high dimension, it is possible for a subgroup of order 4 to deny as many as $4(n+1)$ factors. This can be achieved in E_{64} , where the subgroup $(u^6, u^3v^3, v^3u^3, v^6)$ does deny 28 elements (out of 64). Our method of study for this problem will be to determine how many denied elements can be shared by a pair of generators from a subgroup.

Consider a general subgroup G of order 4:

$$G = (1, a, b, ab) \quad (8)$$

where for notational convenience we have let 1 represent u^4 , and a , b and ab represent the other elements of R_n appearing in G . A detailed examination of the example with G_{19} would have shown an apparent pairing of denied elements by members of a subgroup; for example, uv^3 and vuv^2 both deny u^2v^2 as an element for a coset which is to be disjoint with B and both also deny v^4 . This is no coincidence as the following theorem shows:

Theorem 7.

If two elements, a and b , of a subgroup G , when multiplied by a common factor x , both give products which are in B

$$\begin{aligned}
&\text{and} \quad \left. \begin{aligned} ax &\in B \\ bx &\in B \end{aligned} \right\} \quad (9)
\end{aligned}$$

then there exists another factor x' (which we shall call the conjugate of x) which also satisfies Equation (9)

$$\begin{aligned}
&\text{and} \quad \left. \begin{aligned} ax' &\in B \\ bx' &\in B \end{aligned} \right\}
\end{aligned}$$

Proof: First we note that the elements of B are either u^n or else of the form $u^n - i v u^{i-1}$. Therefore if $a \in B$ and $bx \in B$, there are two possibilities.

Case I

The products differ in two factors:

$$\begin{aligned}
ax &= uu \dots (v)_i \dots (u)_j \dots uu \\
bx &= uu \dots (u)_i \dots (v)_j \dots uu
\end{aligned} \quad (10)$$

where $i \neq j$ since $a \neq b$. Equation (10) shows that a and b must differ in exactly two factors (u and v), and that consequently ab differs from ax and bx in one position for each case, and that ab has two v 's in it

$$ab = uu \dots (v)_i \dots (v)_j \dots uu \quad (11)$$

Furthermore, x must differ from a in the i -th position, and from b in the j -th position;* i.e., if x is given by:

$$x = z_1 z_2 \dots a'_i \dots b'_j \dots z_{n-1} z_n$$

then a and b are of the form:

$$\begin{aligned}
a &= z_1 z_2 \dots a_i \dots b'_j \dots z_{n-1} z_n \\
b &= z_1 z_2 \dots a'_i \dots b_j \dots z_{n-1} z_n
\end{aligned} \quad (12)$$

Let x' be defined to be the element:

$$x' = z_1 z_2 \dots a_i \dots b_j \dots z_{n-1} z_n$$

then

$$\begin{aligned}
ax' &= uu \dots (u)_i \dots (v)_j \dots uu \\
bx' &= uu \dots (v)_i \dots (u)_j \dots uu
\end{aligned}$$

which proves the existence of the conjugate, x' , by construction.

* In this case $'$ means u replaced by v or v replaced by u ; not the usual complement.

Case II

The products differ in only one factor:

$$\begin{aligned} ax &= uu \cdots (u) \cdots uu \\ bx &= uu \cdots (v) \cdots uu \end{aligned} \quad (13)$$

then a and b differ in only the i -th position, and we have by the same construction used above

$$\begin{aligned} ab &= u u \cdots v \cdots u u \\ x &= z_1 z_2 \cdots b_i' \cdots z_{n-1} z_n \\ a &= z_1 z_2 \cdots b_i' \cdots z_{n-1} z_n \\ b &= z_1 z_2 \cdots b_i \cdots z_{n-1} z_n \end{aligned} \quad (14)$$

The conjugate in this case is

$$x' = z_1 z_2 \cdots b_i \cdots z_{n-1} z_n$$

and we have the pair of identities

$$\begin{aligned} ax &= bx' \\ bx &= ax' \end{aligned} \quad (15)$$

which proves the theorem. ■

There are several corollaries based on the argument used in proof of Theorem 7. These form the basis of the next several theorems.

Corollary 1.

If a and b are both elements of B , $a \neq b$, then $ab \notin B$. This follows from Case I above since ab must have two v 's in it, and hence is not in B .

Corollary 2.

If a and b are neither in B , it is still possible for $ab \in B$. Case II gives the necessary condition for this to be true, i.e., that a and b differ in exactly one position. An example would be

$$\begin{aligned} a &= uv^2u \notin B \\ b &= uv^3 \notin B \\ ab &= u^3v \in B \end{aligned}$$

Corollary 3.

It is possible for none of a , b , or ab to be in B . An example is

$$\begin{aligned} a &= u^2v^2 \\ b &= uvuv \\ ab &= uv^2v \end{aligned}$$

Corollary 4.

It is not possible for a and b , $a \neq b$, to share more than a single pair of denied elements. This follows immediately from the Equation (12) and (14) which specify a and b .

We define a distance function on the members of R_n similar to the Hamming distance. The Hamming distance [4] between two binary code words is defined to be the number of positions in which the code words differ. Since all of the elements of R_n can be represented as Kronecker products of factors u and v , we define $d(a,b)$ where $a, b \in R_n$, to be the number of factors which

differ between a and b . For example, if $a = u^2v$ and $b = uvu$, $d(a,b) = 2$. It is easy to show that $d(a,b)$ is a distance function.

Theorem 8.

If $a, b \in G$, $G \subseteq R_n$, and $d(a,b) = k$, then for any $x \in G$

$$d(ax, bx) = k$$

Proof: Theorem 8 is a consequence of u and v being elements of a group so that if in the i -th position a and b were alike (unlike) then ax and bx are also alike (unlike) in the same position. ■

There can only be $2^n - 1$ independent distances in a subgroup of R_n with n generators in view of Theorem 8. For example, in $G = \{1, a, b, ab\}$

$$\begin{aligned} d(a, ab) &= d(1, b) \\ d(b, ab) &= d(1, a) \\ d(a, b) &= d(1, ab) \end{aligned}$$

The vertex 1, i.e., u^0 , will be chosen as the reference point because it must be present in all subgroups.

Theorem 9.

There exists an element $x \in G$, such that $d(1, x) = 1$ if and only if the other two elements y and xy satisfy the pair of identities:

$$\begin{aligned} d(1, y) &= k \\ d(1, xy) &= k+1 \end{aligned} \quad (16)$$

Proof: Since $d(1, x) = 1$, $x \in \{B - u^n\}$, i.e., $x = u^{n-1}vu^{i-1}$. Therefore if

$$y = z_1 z_2 \cdots z_i \cdots z_{n-1} z_n \quad (17)$$

then

$$xy = z_1 z_2 \cdots z_i' \cdots z_{n-1} z_n \quad (18)$$

If $z_i = u$, then $d(1, xy) = d(1, y) + 1$, since xy will differ in one more position than x from u^n . Similarly, if $z_i = v$ then $d(1, xy) = d(1, y) - 1$ which can be put in proper form by selecting the term with u in the i -th position to be y .

On the other hand if y and xy are of the form given in Equations (17) and (18) then $x = y(xy)$ will have all factors a except the i -th factor, and hence $x \in \{B - u^n\}$ and $d(1, x) = 1$. ■

Corollary 1.

If $a, b \in G$ and $a, b \in B$, then $ab \notin B$. This was proven earlier but is trivially apparent from the result of Theorem 9. In particular, $d(1, ab) = 2$.

Theorem 10.

There exists an element $x \in G$, such that $d(1, x) = 2 \leq d(1, g)$ for all $g \in G$, if and only if the other two elements y and xy satisfy one of the two pairs of identities

$$\begin{aligned} d(1, y) &= k \\ d(1, xy) &= k \end{aligned} \quad (19)$$

or

$$\begin{aligned} d(1, y) &= k \\ d(1, xy) &= k+2 \end{aligned} \quad (20)$$

Proof: Since $d(1, x) = 2$, we know that x has exactly two v factors in it:

$$x = u^{n-i-j} v u^{j-1} v u^{i-1}$$

Therefore if

$$y = z_1 z_2 \dots z_{i+j} \dots z_i \dots z_{n-1} z_n \quad (21)$$

then

$$xy = z_1 z_2 \dots z_{i+j}' \dots z_i' \dots z_{n-1} z_n \quad (22)$$

There are two possibilities: either z_{i+j} and z_i are both u , or one is u and one is v . First assume they are both u , then

$$d(1, xy) = d(1, y) + 2$$

and Equations (20) are proven. If one is u and one is v , then

$$d(1, xy) = d(1, y)$$

and Equations (19) are proven. By the same argument used in Theorem 9 we must select the proper term to be called y for the above to be true. If y and xy are of the form given in Equations (21) and (22), irrespective of whether z_i and z_{i+j} are u or v , the product is of the desired form for x . ■

Corollary 1.

There are subgroups $G = \langle 1, a, b, ab \rangle$ such that

$$d(1, a) = d(1, b) = d(1, ab) = 2 \quad (23)$$

Discussion:

Equation (23) is true by Theorem 10 for all $n \geq 3$, however it is also possible to define the precise form which the vectors a, b and ab must have for Equation (23) to be satisfied. If y and xy are to each be at a distance of two from u^n , then x, y and xy are each made up of factors of u in all but three positions in which the factors are

$$\begin{array}{ll} a & u_1 \dots v_j \dots v_k \\ b & v_1 \dots v_j \dots u_k \\ ab & v_1 \dots u_j \dots v_k \end{array}$$

All of the cases of interest are tabulated below, as determined from Theorems 9 and 10 and simple combinatorial considerations.

$$\begin{array}{ll} \text{Case I} & d(1, a) = 1 \\ & d(1, b) = 1 \\ & d(1, ab) = 2 \end{array}$$

The number of such subgroups is:

$$N_1 = \binom{2^n}{2} = 2^{n-1}(2^n - 1) \quad (24)$$

$$\begin{array}{ll} \text{Case II} & d(1, a) = 1 \\ & d(1, b) = 2 \\ & d(1, ab) = 3 \end{array}$$

By Theorem 8 $d(1, b) = 2$ implies that $d(a, ab) = 2$. But $d(1, a) = 1$, therefore

$$a = uu \dots u_1 \dots u_j \dots v_k \dots uu$$

and

$$b = uu \dots v_1 \dots v_j \dots u_k \dots uu$$

$$ab = uu \dots v_1 \dots v_j \dots v_k \dots uu$$

The vector a can be selected in 2^n ways and any

pair of the unoccupied positions can be used for the v entries in b , therefore the number of such subgroups is:

$$N_2 = 2^n \binom{2^n - 1}{2} \quad (25)$$

$$\begin{array}{ll} \text{Case III} & d(1, a) = 1 \\ & d(1, b) = k \\ & d(1, ab) = k+1 \end{array}$$

where $k > 2$. The number of subgroups which belong to Case III is:

$$N_3 = 2^n (2^{n-1} - 1) (2^{n-1} - 1) \quad (26)$$

since we can select an element of $\{B - u^n\}$ in 2^n ways and we can then select an element with two v 's in it, neither of which coincides in position with the v in a , in $(2^{n-1} - 1)(2^{n-1} - 1)$ ways.

$$\begin{array}{ll} \text{Case IV} & d(1, a) = 2 \\ & d(1, b) = 2 \\ & d(1, ab) = 2 \end{array}$$

Corollary 1 to Theorem 10 shows that the number of subgroups of this type is just the number of ways in which three positions can be selected, i.e.,

$$N_4 = \binom{2^n}{3} \quad (27)$$

$$\begin{array}{ll} \text{Case V} & d(1, a) = 2 \\ & d(1, b) = 2 \\ & d(1, ab) = 4 \end{array}$$

Theorem 10 showed that z_i and z_{i+j} in Equations (21) and (22) must both be u in this case, therefore:

$$N_5 = \frac{1}{2} \binom{2^n}{2} \binom{2^n - 2}{2} \quad (28)$$

$$\begin{array}{ll} \text{Case VI} & d(1, a) = 2 \\ & d(1, b) = k \\ & d(1, ab) = k+2 \end{array}$$

where $k > 2$

$$N_6 = \binom{2^n}{2} \sum_{k=3}^{2^n - 2} \binom{2^n - k}{k} \quad (29)$$

$$\begin{array}{ll} \text{Case VII} & d(1, a) = 2 \\ & d(1, b) = k \\ & d(1, ab) = k \end{array}$$

where $k > 2$

$$N_7 = \binom{2^n}{2} \sum_{k=3}^{2^n - 2} \binom{2^n - k}{k-1} \quad (30)$$

$$\begin{array}{ll} \text{Case VIII} & d(1, a) = d_1 \\ & d(1, b) = d_2 \\ & d(1, ab) = d_3 \end{array}$$

where

$$d_i > 2$$

If $n < 6$, there are no examples of Case VIII. The smallest such example is:

a = uuuvvv
b = vvvuvu
ab = vvvvv

The simplest way to express the number of subgroups belonging to Case VIII is

$$N_G = \frac{(2^n-1)(2^{n-1}-1)}{3} - \sum_{i=1}^7 N_i \quad (31)$$

The classification of the subgroups of R_n of order 4 into the eight cases given above may appear arbitrary, but is based on how the elements within the subgroups share the elements denied for use in the construction of cosets disjoint from B. We can now say just how many elements an arbitrary subgroup (of order 4) denies.

Lemma: If $a \in B$, then a and I jointly deny 2^n elements.

Proof: Theorem 7 applies since $aa = I$, and the number of denied positions is

$$2(n+1) - 2 = 2n. \blacksquare$$

Theorem 11.

If G is a subgroup of type i, then the number of elements denied by G is n_i

i	n_i
I	$4(n-1)$
II	$4(n-1)$
III	$4n$
IV	$4(n-1)$
V	$4(n-1)$
VI	$4n$
VII	$4n$
VIII	$4(n+1)$

Proof: Each case is proven by forming the multiplication for G with R_n to determine the shared denied elements using the results of the last few theorems. \blacksquare

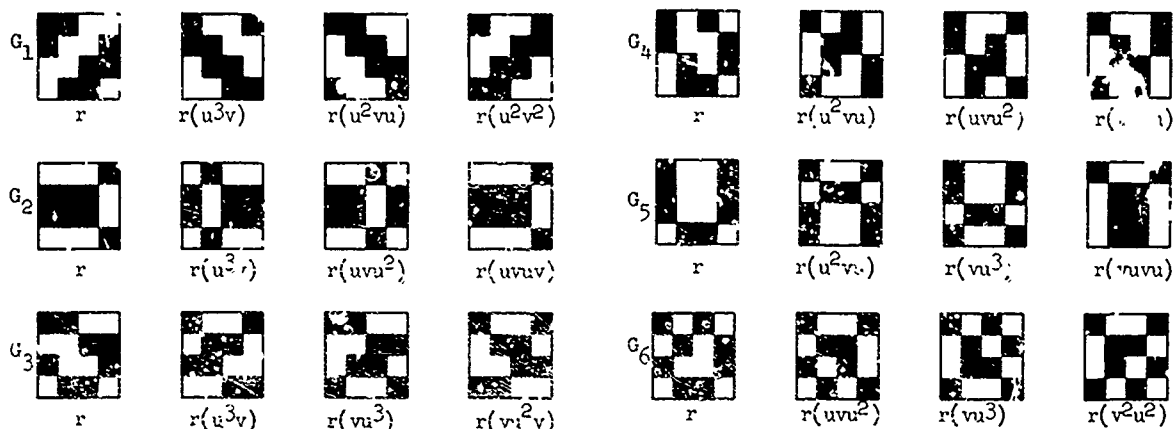
In view of Theorem 11 only the subgroups of Types I, II, IV and V can possibly have cosets disjoint from B_n in R_n . Since the elimination procedure which we have developed in the preceding pages actually eliminated coset images, each of the sets of four undenied elements (for $n=4$) is itself a coset. The corresponding replacement vectors, generated using Theorem 4, are shown in Figure 1 each of which generates a non-equivalent (to R_n) basis.

Conclusion

A general technique, dependent on the subgroup structure of R_n considered as a multiplicative involutory group, has been developed which can be used to construct inequivalent bases for E_{2n} . The theory was applied in the second half of the paper to a complete enumeration of the bases obtainable in this manner for E_{16} . The method is applicable to all R_n , $n \geq 4$. Limitations of space precluded a discussion of the transformation groups of the various bases, however, it should be noted that the invariants of patterns are in general changed by the coordinate transformations given here.

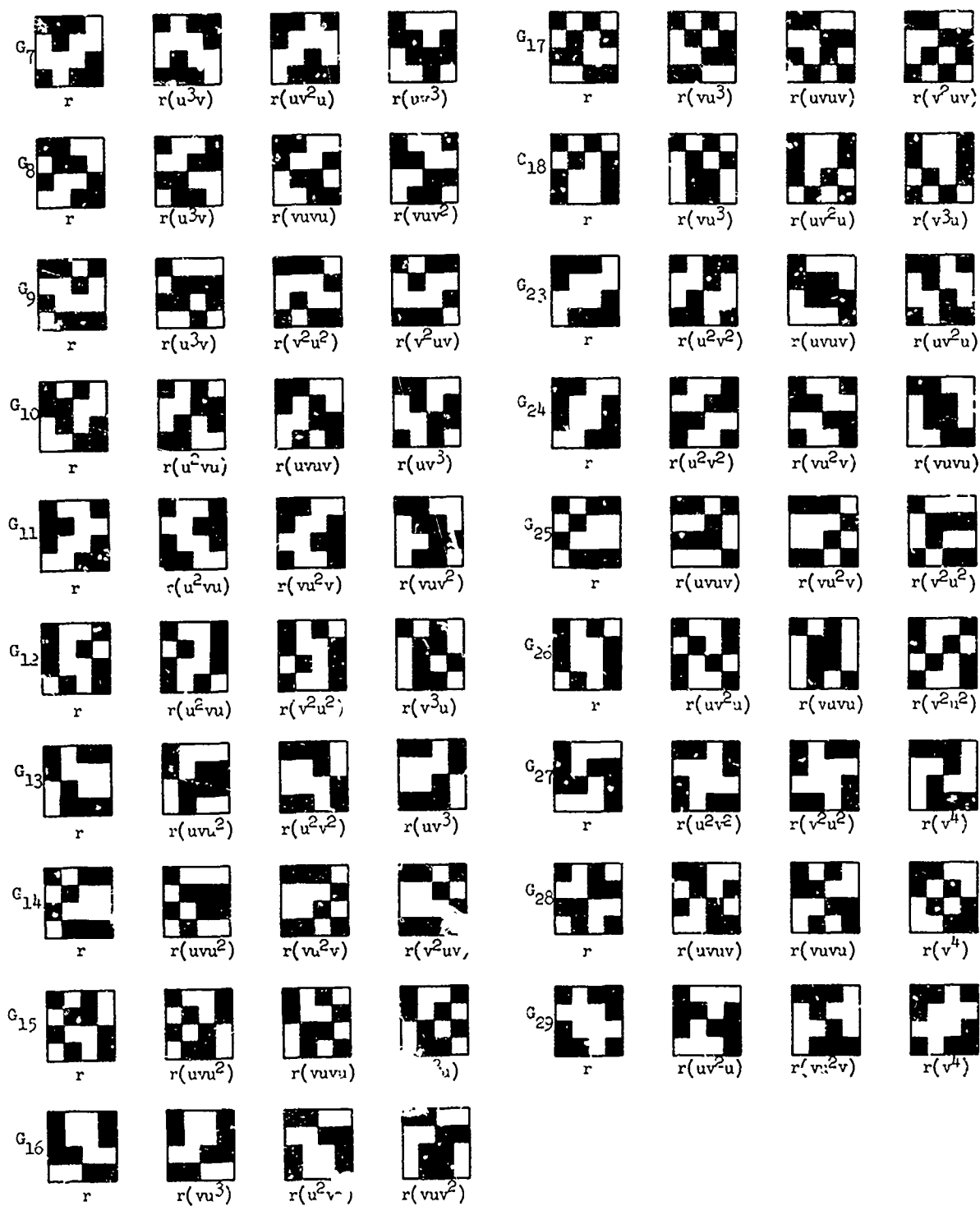
References

- [1] Marshall Hall, Combinatorial Theory, Blaisdell Publishing Co., 1967, pp. 204-222.
- [2] von Hans Dieter Lüke, Lineare Signalverknüpfung in der Multiplextechnik, Archiv der Elektrischen Übertragung, Volume 24, No. 2, February 1970, pp. 57-65.
- [3] R. D. Carmichael, Introduction to the Theory of Groups of Finite Order, Dover Publications Inc., New York, 1956, Chapter IV.
- [4] R. W. Hamming, Error Detecting and Error Correcting Codes, Bell System Technical Journal, Volume 29, 1950, pp. 147-160.



Replacement vectors in h_4

Figure 1 (cont.)



Replacement vectors in R_4

Figure 1

A NEW METHOD FOR REPRESENTING WALSH FUNCTIONS

Ira Ross

John J. Kelly
Associate Professor of Electrical Engineering
New York University
Bronx, New York 10453
U.S.A.

Abstract

A closed form representation for the set of Walsh functions is presented, in terms of the sign of a product expansion of sines and cosines. This provides a clear picture of the Walsh functions and their relationship to trigonometric functions. Trigonometry identities show the relationships between the Walsh functions and a simple procedure gives the Walsh function of any argument.

Introduction

A complete set of orthogonal functions, taking only the values ± 1 were presented by Walsh¹ in 1923. The functions can be defined iteratively or as products of the Rademacher functions, a subset of the Walsh functions. Herein is presented a general closed form representation of Walsh functions in terms of the sign of a product expansion of sines and cosines. The product form presented makes periodic continuation of the Walsh functions easy to visualize, graph, and manipulate.

Trigonometric Expression of Walsh Functions

Consider the set of Walsh functions, $\{wal(j, x)\}$, where j indicates the Walsh function index and x is the argument. They form a complete set, orthogonal in the interval $-\frac{1}{2} \leq x < \frac{1}{2}$ and zero elsewhere. This normalized interval can be converted to the general interval $-T/2 \leq x < T/2$ with x replaced by x/T , as required. The general expression for $wal(j, x)$ is based on binary representation of the index j , i.e.,

$$j = j_n 2^n + j_{n-1} 2^{n-1} + \dots + j_1 2^1 + j_0 2^0 \quad (1)$$

where $j_i = 0$ or 1 , so that

$$wal(j, x) = \text{sign}[(\sin 2\pi x)^{j_0} \prod_{k=1}^n (\cos 2^k \pi x)^{j_k}] \quad (2)$$

The sign function is similar to the signum function except that it takes the value zero only when the limit from the right is not equal to the limit from the left.

The even and odd Walsh functions of $cal(i, x)$ and $sal(i, x)$ are easily found from (2) using the relations $cal(i, x) = wal(2i, x)$ and $sal(i, x) = wal(2i-1, x)$ and writing the indices $2i$ and $2i-1$ as binary numbers as in (1). We note that

$$cal(i, x) = \text{sign}[\prod_{k=0}^n (\cos 2^{k+1} \pi x)^{i_k}] \quad (3)$$

To express $sal(i, x)$ in terms of index i we must specify the binary representation of i as follows. Let the lowest binary coefficient of i , not equal to zero, be i_s , so that $i_s = 1$.

$i_s - 1 = i_s - 2 = \dots = i_0 = 0$. Then by direct substitution in (2) and combining terms

$$sal(i, x) = \text{sign}[(\sin 2^{s+1} \pi x)^n \prod_{k=s+1}^n (\cos 2^{k+1} \pi x)^{i_k}] \quad (4)$$

Equations (2), (3), and (4) completely define the Walsh functions, ordered by sequence, orthogonal in the interval $-\frac{1}{2} \leq x < \frac{1}{2}$ and also periodically continued from this interval over the real axis.

Example

As an example consider the construction of the function $wal(9, x) = sal(5, x)$. In binary form $j = 9 = 1001$ $i = 5 = 101$. Then

$$wal(9, x) = sal(5, x) = \text{sign}[\sin 2\pi x \cos 8\pi x]$$

which is illustrated in Figure 1.

The general expression given in (2) is proved by mathematical induction using Harmuth's² iterative difference equation for the Walsh functions. This is

$$wal(2j+p, x) = (-1)^{j/2+p} \{wal[j, 2(x + \frac{1}{4})] + (-1)^{j+p} wal[j, 2(x - \frac{1}{4})]\} \quad (5)$$

where $p = 0$ or 1 $j = 0, 1, 2, \dots$ [$j/2 =$ integer value and $wal(0, x) = 1$ for $-\frac{1}{2} \leq x < \frac{1}{2}$]

$$wal(0, x) = 0 \text{ for } x < -\frac{1}{2}, x > \frac{1}{2}$$

Equation (5) is separated into two parts, one valid for $-\frac{1}{2} \leq x < 0$ and the other for $0 < x \leq \frac{1}{2}$. The proof entails four separate cases, corresponding to $p = 0, 1$ and $j_0 = 0, 1$, as stated. Direct substitution of (2) into (5) under the above conditions proves that

$$wal(2j+p, x) = \text{sign}[(\sin 2\pi x)^p \prod_{k=0}^n (\cos 2^{k+1} \pi x)^{j_k}] \quad (6)$$

which is the required result for the induction proof

Application

Some basic properties of the Walsh functions are evident from (2), such as $wal(j, x)$ is an even function of x when j is even (equal to $cal(j/2, x)$), and an odd function of x when j is odd (equal to $sal(\frac{j+1}{2}, x)$). Also the relationships between the even and odd Walsh functions are

$$wal(2i+1, x) = wal(-i, x); x > 0 \\ = -wal(2i, x); x < 0 \quad (7)$$

$$\text{and } sal(i+1, x) = cal(i, x); x > 0 \\ = -cal(i, x); x < 0$$

The value of i , satisfying the following periodic relationships of the Walsh functions

can easily be found from (3) and (4).

$$\text{cal}(i, x+x_0) = \text{sal}(i, x) \quad (8a)$$

$$\text{sal}(i, x+x'_0) = \text{cal}(i, x) \quad (8b)$$

$$\text{cal}(i, x+x_1) = -\text{cal}(i, x) \quad (8c)$$

$$\text{sal}(i, x+x'_1) = -\text{sal}(i, x) \quad (8d)$$

$$\text{cal}(i, x+x_2) = \text{cal}(i, x) \quad (8e)$$

$$\text{sal}(i, x+x'_2) = \text{sal}(i, x) \quad (8f)$$

For example consider relation (8a) when the lowest binary coefficient of i not equal to zero is i_s . For this value of i substituted into (3) we obtain

$$\begin{aligned} \text{cal}(i, x \pm |x_0|) \\ = \text{sign} \left\{ \prod_{k=s}^n [\cos 2^{k+1} \pi (x \pm |x_0|)]^{i_k} \right\} \end{aligned} \quad (9)$$

We must determine the sign and magnitude of x_0 , which depends on s , so that (9) becomes

$$\begin{aligned} \text{cal}(i, x) \\ = \text{sign}[(\sin 2^{s+1} \pi x) \prod_{k=s+1}^n (\cos 2^{k+1} \pi x)^{i_k}] \end{aligned} \quad (10)$$

This can be done by equating the product expansions in (9) and (10) term by term. Equating the first terms, i.e., $k=s$ in (9) and $\sin 2^{s+1} \pi x$ shows that $|x_0| = 2^{-s-2}$ is the smallest $|x_0|$ satisfying the equality and is analogous to the principle value in trigonometric functions. Substituting this value of x_0 into (9) yields the equation

$$\begin{aligned} \text{cal}(i, x \pm |x_0|) \\ = \text{sign} \{ (\pm 1)(\sin 2^{s+1} \pi x) (-1)^{i_{s+1}} \prod_{k=s+1}^n (\cos 2^{k+1} \pi x)^{i_k} \} \end{aligned} \quad (11)$$

To make (10) and (11) equal we see that the sign of x_0 must be chosen according to the value of i_{s+1} , i.e., if

$$\begin{aligned} i_{s+1}=1, \text{ then } x_0 = +2^{-s-2} \\ i_{s+1}=0, \text{ then } x_0 = -2^{-s-2} \end{aligned} \quad (12)$$

This relationship is equivalent to the one described by Harmuth.² A similar development for (8b) yields the relationships

$$\begin{aligned} \text{if } i_{s+1}=1, \text{ then } x'_0 = -2^{-s-2} \\ \text{and if } i_{s+1}=0, \text{ then } x'_0 = +2^{-s-2} \end{aligned} \quad (13)$$

For (8c) to (8f), the relations are

$$x_1 = x'_1 = \pm 2^{-s-1}, \text{ and } x_2 = x'_2 = \pm 2^{-s}.$$

In general, any desired periodic relationship for the Walsh functions can be found using the product expansion described in the paper.

The product rule for multiplication of Walsh functions is demonstrated using (2). The product rule is

$$\text{wal}(j, x) \text{wal}(h, x) = \text{wal}(j \oplus h, x) \quad (14)$$

From (2) we obtain

$$\begin{aligned} \text{wal}(j, x) \text{wal}(h, x) \\ = \text{sign}[(\sin 2\pi x)^{j_0} \prod_{k=1}^n (\cos 2^k \pi x)^{j_k}] \\ \cdot \text{sign}[(\sin 2\pi x)^{h_0} \prod_{k=1}^n (\cos 2^k \pi x)^{h_k}] \end{aligned} \quad (15)$$

where $n \geq n'$.

Now we may combine the right hand side and obtain

$$\begin{aligned} \text{wal}(j, x) \text{wal}(h, x) \\ = \text{sign}[\sin 2\pi x)^{j_0+h_0} \prod_{k=1}^n (\cos 2^k \pi x)^{j_k+h_k}] \end{aligned} \quad (16)$$

When $j_i \neq h_i$ the term in the product expansion is raised to the first power and remains. When $j_i = h_i = 1$ the term is squared and is always positive so that it can be removed from the expansion. This operation is equivalent to the modulo 2 addition of j and h so that the product rule is demonstrated.

Another application of the results presented leads to a Walsh function generator similar to the one proposed by Davies³ but using the subset $\text{wal}(2^k, x)$ to generate the Walsh functions instead of the subset $\text{wal}(2^{k+1}-1, x)$ (the Rademacher functions). Examination of (2) shows that the Walsh functions $\text{wal}(2^k, x)$, $k=0, 1, \dots$ can be represented by

$$\begin{aligned} \text{wal}(2^0, x) &= \text{sign}[\sin 2\pi x], k=0 \\ \text{wal}(2^k, x) &= \text{sign}[\cos 2^k \pi x], k \neq 0 \end{aligned} \quad (17)$$

This set of functions can be generated as easily as the set of Rademacher functions

$$\text{wal}(2^{k+1}-1, x) = R_k(x) = \text{sign}[\sin 2^{k+1} \pi x]. \quad (18)$$

The individual functions used to generate a particular Walsh function are found directly from the binary representation of the index j without resorting to the Gray code transformation of j as in Davies' scheme. Thus

$$\text{wal}(j, x) = \prod_{k=0}^n [\text{wal}(2^k, x)]^{j_k} \quad (19)$$

The advantage of using (19) is not significant when zero crossings are not considered, i.e., when Walsh functions are generated digitally. However, when the continuous Walsh functions are needed in analog form the use of the Rademacher functions to generate them presents a problem. This problem will always arise when any product of odd Walsh functions is used to generate an even Walsh function. The odd Walsh functions have zero crossings at points where the even functions do not. This is evident at $x=0$ where all the odd functions have a zero crossing and the even functions are always positive one. The set $\{\text{wal}(2^k, x)\}$ of (19) generates the functions without error.

Conclusions

The representation of the Walsh functions in terms of the sign of a product expansion of

sines and cosines provides a convenient closed form for the Walsh functions. Many basic properties of the functions are clearly evident from the product expansion. This approach also simplifies computation of the Walsh functions. It was shown that the subset of Walsh functions, $wal(2^k, x)$, can be easily generated and used to generate the complete set of Walsh functions without zero crossing error.

Bibliography

1. Walsh, J.L., "A Closed Set of Orthogonal Functions," Amer. J. of Mathematics, (S5) pp. 5-24, 1923.
2. Harmuth, H.F., "Transmission of Information by Orthogonal Function," New York: Springer, 1969.
3. Davies, A.C., "On the Definition and Generation of Walsh Function," IEEE Transactions on Computers, Vol. C-21, No. 2, pp. 187-189, February 1972.

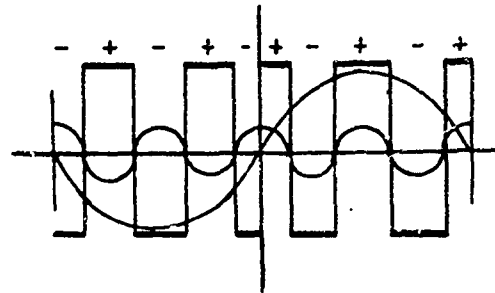


FIG. 1. GENERATION OF $wal(9, x)$
From $\text{sign}[(\sin 2\pi x)(\cos 8\pi x)]$

ON WALSH DESCRIBING FUNCTIONS

Chon Tam Le Dinh, graduate student
Roger Goulet, assistant professor
Noel Van Houtte, assistant professor
Department of Electrical Engineering
University of Sherbrooke, Sherbrooke, Canada

Abstract

Important elements in dealing with linear control systems are stability and performance and there are many methods to evaluate these factors. Nonlinear systems are more difficult to handle. Stability can be looked at by means of analytical methods without restrictions on the input signal. Performance for design purposes can be handled using approximations on the input signal.

The classical method to study nonlinear systems for system design is the describing function method. It requires that the form of the input signal to derive this function approximates the actual signal at the nonlinearity input. This input signal is taken to be a sinusoid.

The present paper studies the describing function using a Walsh function input, which may take the form of a single function up to a sum of several inputs. After the definition of the Walsh describing function, some fundamental remarks are given, based on the properties of Walsh functions. Next, particular attention is given to a special class of nonlinearities: the odd nonlinearity, after which a general formula is set up for a finite sum describing function.

Finally, the describing function is evaluated for a set of common nonlinearities, which makes it useful for ready computation. It is also pointed out that the advantages due to Walsh functions disappear for nonlinearities with memory, because of the multiple valued definition.

It is concluded that the Walsh describing functions are extremely useful in the study of systems, which mainly consist of nonlinearities especially of the relay type, and that its very nature may yield a better means for identification of such nonlinear elements for modelling purposes.

INTRODUCTION

1. Formulation of the Describing Function Method for Zero-Memory Non Linearity [1]

Consider a single memoryless nonlinear elements of the following form:

$$y = f(x) \quad (1)$$

Assuming that the input $x(t)$ is of a specific form $x_s(t)$, we wish to seek an equivalent set of gains which yields:

$$y_s(t) = N^T(x_s)x_s(t) \quad (2)$$

where the equivalent gain column matrix N is chosen such as to minimize the mean-squared error between the actual output $f(x_s)$ and the approximate output $y_s(t)$. Thus if, with fig.1:

$$e(t) \triangleq f(x_s) - N^T(x_s)x_s \quad (3)$$

then $N(x_s)$ is to be chosen such as to minimize $\|e\|^2$:

$$\begin{aligned} \|e(t)\|^2 &= \lim_{\tau \rightarrow \infty} \frac{1}{2\tau} \int_{-\tau}^{\tau} |e(t)|^2 dt \\ &= \overline{f^2} - 2\overline{N^T x_s f} + (\overline{N^T x_s})^2 \end{aligned} \quad (4)$$

The minimization of $\|e\|^2$ with respect to N_j can be carried out by setting $\partial \|e\|^2 / \partial N_j$ equal to zero. This yields

$$\sum_{k=1}^n N_k x_{s_k} x_{s_j} = \overline{f x_{s_j}} \quad \text{for } j=1,2,\dots \quad (5)$$

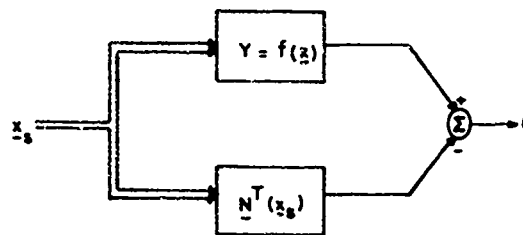


Fig. 1 - Block Diagram Representation of The Describing Function and The Output Error.

2. One Input - One Output

For this case, the expression (5) becomes

$$N = \overline{f x_s} / \overline{x_s^2} \quad (6)$$

We now consider x_s to be a Walsh function input, (notation as in [2])

$$x_s(t) = \pm 1(k, t/T) ; k=0,1,2,\dots \quad (7)$$

where $T = 1/\Delta$ is the period (or the finite definition time) of $x_s(t)$.

In order to evaluate N from (6), we have

$$\overline{f x_s} = (a/2^n) \int_0^{2^n} f[a\phi(k, t/T)] \phi(k, t/T) dt$$

By substituting $v = t/\Delta$ the latter formula becomes

$$\overline{f x_s} = (a/2^n) \int_0^{2^n} f[a\phi(k, v2^{-n})] \phi(k, v2^{-n}) dv.$$

But the function $\phi(k, v2^{-n})$ is constant in each interval $n < v < u+1$ where $u = 0, 1, 2, \dots, 2^n-1$.

$$\overline{f x_s} = a2^{-n} \sum_{u=0}^{2^n-1} f[a\phi(k, u2^{-n})] \phi(k, u2^{-n}). \quad (8)$$

On the other hand:

$$\begin{aligned} \overline{x_s^2} &= (a^2 2^{-n}) \int_0^{2^n} \phi^2(k, v2^{-n}) dv \\ &= a^2 2^{-n} \sum_{n=0}^{2^n-1} \phi^2(k, u2^{-n}) = a^2 \end{aligned} \quad (9)$$

Finally, the describing function (6), for a single Walsh function input, is

$$N(a) = (1/2^n a) \sum_{u=0}^{2^n-1} f[a\phi(k, u2^{-n})] \phi(k, u2^{-n}) \quad (10)$$

Later we will show that the second member of (10) is independent of k .

3. Multiple Input - One Output

A generalization of the single input case is one where we may have a combination of two or more inputs, which are a Walsh function, and where one observes one output.

Let the input be x_s of the period T (or defined for $0 < t < T$):

$$x_s = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_m \end{bmatrix} \quad \text{where } x_i = a_i \phi(k_i, t/T). \\ k_i \neq k_j \quad (i \neq j)$$

while $y = f(x_1, x_2, \dots, x_m)$

$$y_s = [N_1 \ N_2 \ \dots \ N_m] x_s. \quad (11)$$

We remark that x_i and x_j are two orthogonal functions, $\overline{x_i x_j} = 0$ for $i \neq j$, and from (5)

$$N_i = \overline{x_i f} / \overline{x_i^2}; \quad i = 1, 2, \dots, m \quad (12)$$

The mean-squared error is so equal to

$$|e|_{\min}^2 = \overline{f^2} - \sum_{i=1}^m N_i^2 \overline{x_i^2}. \quad (13)$$

FUNDAMENTAL REMARKS FOR COMPUTATION OF WALSH DESCRIBING FUNCTIONS

The computation of the Walsh input describing functions (WIDF) (10) and/or (12) is

relatively easy with and without a computer because the discreteness of Walsh functions:

From (10) and (12) we have

$$N(a) = (1/2^n a) \sum_{u=0}^{2^n-1} f[awal(k, u)] wal(k, u) \quad (14)$$

and

$$N_i(a_1, \dots, a_m) = (1/2^n a_i) \sum_{u=0}^{2^n-1} f[x_1, \dots, x_m] wal(k_i, u) \quad (15)$$

where $i = 1, 2, \dots, m$

$$k_i = \sum_{r=0}^{n-1} k_{i,r} 2^r; \quad k_{i,r} \in \{0, 1\} \quad (16)$$

$$wal(k_i, u) = \phi(k_i, u2^{-n})$$

$$\text{and } [3] \ wal(k_i, u) = -1^S \quad (17)$$

$$\text{with } S = \{k_{i,n-1} u_0 \oplus \sum_{r=1}^{n-1} (k_{i,n-r} \oplus k_{i,n-r-1}) u_n\}$$

where \oplus means the dyadic sum and \sum logical sum.

1. Theorem 1 - Sequency-independent WIDF

The second member of (14) is independent of $k \neq 0$, and

$$N(a) = (1/2a) [f(a) - f(-a)] \quad (18)$$

Proof: Consider the summation in (14).

$\forall k \neq 0$, we have

2^{n-1} terms $f(a)$ corresponding to $wal(k, u) = 1$
and
 2^{n-1} terms $f(-a)$ corresponding to $wal(k, u) = -1$.

So the summation is independent of k , and (18) is evident. QED.

Remarks 1:

The following conclusions can be drawn from the last formula:

R1-1. No computation is necessary for the one input case.

R1-2. If $f(x)$ is even, $N(a) = 0$ for $k \neq 0$

R1-3. If $f(x)$ is odd,

$$N(a) = (1/a) f(a), \quad (19)$$

and particularly for this case the approximation error ϵ (3) is identically zero. This is not true for the sinusoidal DF.

In fact from (3) and (19) we get

$$\begin{aligned} \epsilon &= f[awal(k, u)] - f(a) wal(k, u) \\ &= \begin{cases} f(a) - f(a), & \text{if } wal(k, u) = 1 \\ f(-a) + f(a), & \text{if } wal(k, u) = -1 \end{cases} \end{aligned}$$

$$\text{Hence } \epsilon = 0 \quad (20)$$

R1-4. $N(a)$ is always real.

R1-5. For an unbalanced non linearity (18) gives a constant error; but it is not difficult to get this bias term.

2. Theorem 2 - Sequence Independent TWIDF (Two WIDF)

*If the zero-mean input is

$$\underline{x}(u) = [a_1 \text{wal}(k_1, u) \ a_2 \text{wal}(k_2, u)]^T; k_1, k_2 \neq 0 \quad (21)$$

then the second member of (15) is independent of k_1 and k_2 :

$$N_1(a_1, a_2) = \frac{1}{4a_1} [f(a_1, a_2) + f(a_1, -a_2) - f(-a_1, a_2) - f(-a_1, -a_2)] \quad (22)$$

$$N_2(a_1, a_2) = N_1(a_2, a_1) \quad (23)$$

*If the input contains a DC term, i.e.,

$$\underline{x}(u) = [a_0 \text{wal}(0, u) \ a_1 \text{wal}(k_1, u)]^T; k_1 \neq 0 \quad (24)$$

then the second member of (15) is independent of k_1 :

$$N_0(a_0, a_1) = (1/2a_0) [f(a_0, a_1) + f(a_0, -a_1)] \quad (25)$$

$$N_1(a_0, a_1) = (1/2a_1) [f(a_0, a_1) - f(a_0, -a_1)] \quad (26)$$

Proof: Consider the summation of (15).

By the orthogonality of the discrete functions $\text{wal}(k_1, u)$ and $\text{wal}(k_2, u)$ it is clear that: for all $k_1 \neq 0$ and for all $k_2 \neq 0$ we have

2^{n-2} terms corresponding to

$$\begin{array}{cc} \text{wal}(k_1, u) = 1 & , \quad \text{wal}(k_2, u) = +1 \\ & 1 & , \quad -1 \\ & -1 & , \quad 1 \\ & -1 & , \quad -1 \end{array}$$

Consequently,

$$N_2(a_1, a_2) = (1/4a_2) [f(a_1, a_2) + f(-a_1, a_2) - f(-a_1, -a_2) - f(a_1, -a_2)] \quad (27)$$

The comparison between the latter expression and (22) yields (23).

Remarks

R2-1. (18) is a particular case of (22) with $a_2 = 0$. On the other hand it is not difficult to generalize (22). For example, if one has m zero mean inputs ($m \leq n$, of course; we will consider the case $m > n$ in more detail in section 4) then:

$$N_1(a_1, \dots, a_m) = \frac{1}{2^m a_1} \sum_{r_j=0}^1 (-1)^{r_1} f[(-1)^{r_1} a_1, \dots, (-1)^{r_m} a_m] \quad (28)$$

where $i = 1, 2, \dots, m < n$, and 2^n is the period of the set $\{\text{wal}(k_i, u)\}$.

The latter formula is still valid if one or more $a_j (j \neq i)$ are equal to zero.

Note that $N_1(a_1, a_m)$, independent of k_1, \dots, k_m is also independent of the set $\text{wal}(k_i, u)$; $i = 1, \dots, m \leq n$.

R2-2. By the same token if one has a DC term and m zero mean terms ($m \leq n$); one gets:

$$N_0(a_0, \dots, a_m) = \frac{1}{2^m a_0} \sum_{r_j=0}^1 f[a_0, (-1)^{r_1} a_1, \dots, (-1)^{r_m} a_m], \quad (29)$$

$$N_1(a_0, \dots, a_m) = -\frac{1}{2^m a_1} \sum_{r_j=0}^1 (-1)^{r_1} f[a_0, (-1)^{r_1} a_1, \dots, (-1)^{r_m} a_m] \quad (30)$$

where $i = 1, 2, \dots, m \leq n$ and a_0 is the amplitude of the DC term.

Note again that (29) and (30) are independent of the set $\{\text{wal}(k_i, u)$; $i = 1, 2, \dots, m < n\}$.

R2-3. $N_i(a)$, $i = 0, 1, \dots$ are always real.

If m is small these formulas (28) (29) (30) are handy for rapid computation. On the other hand, for large values of m , the evaluation is easy with a computer program.

LINEARIZATION SEPARABILITY

For this section we will consider the input $\underline{x}(u)$ in the following form:

$$\underline{x}(u) = \sum_{i=1}^m a_i \text{wal}(k_i, u) \quad (31)$$

where $0 \leq k_1 < k_2 < \dots < k_m$ and $a_i \neq 0$ for $i = 1, \dots, m$.

1. Definition 1: Strictly Linearization Separability

Let the input be (31). The output $y(u) = f(\underline{x}(u))$ of a nonlinearity is called strictly separable of order m if

$$y(\underline{x}(u)) = y_s(u) = \sum_{i=1}^n N_i(a_1, \dots, a_m) a_i \text{wal}(k_i, u) \quad (32)$$

i.e., $\varepsilon(u) = y(\underline{x}(u)) - y_s(u) = 0$ for all u . (33)

2. Theorem 3 - Linearization Separability of Order 2

*The output of an arbitrary nonlinearity is strictly separable into DC and AC Walsh components, i.e. if the input is

$$\underline{x}(u) = a_0 + a_1 \text{wal}(k, u), \quad k \neq 0 \quad (34)$$

then

$$y(u) = a_0 N_0(a_0, a_1) + a_1 N_1(a_0, a_1) \text{wal}(k, u) \quad (35)$$

*Any odd nonlinearity is strictly separable up to order 2, i.e. $m = 1$ and $m = 2$

Proof:

*Let the input be (34). From (25) and (26) we have for any nonlinearity (not necessarily odd)

$$\begin{aligned} 2a_0N_0(a_0, a_1) &= f(a_0+a_1) + f(a_0-a_1) \\ N_1(a_0, a_1) &= f(a_0+a_1) - f(a_0-a_1) \end{aligned} \quad (36)$$

$N_0(0, u)$ is always constant and equal to 1. Thus from (36) we have the table 1.

In comparing the two last columns we see that $u = 0$.

*We have shown the theorem for $m = 1$ (see R1-3).

For $m = 2$, the 1st part of this theorem has shown the separability between DC and AC Walsh components. Let the input now be

$$x(u) = a_1 \text{wal}(k_1, u) + a_2 \text{wal}(k_2, u). \quad (37)$$

The nonlinearity is odd, $f(x) = f(-x)$. Thus the set of equations (22) and (23) must be written as

$$2a_1N_1(a_1, a_2) = f(a_1+a_2) + f(a_1-a_2) \quad (38)$$

$$2a_2N_2(a_2, a_1) = f(a_1+a_2) - f(a_1-a_2). \quad (39)$$

$f(x)$ is odd, so (39) is equal to

$$2a_2N_2(a_1, a_2) = f(a_1+a_2) - f(a_1-a_2). \quad (40)$$

From (38) and (40) consider the table 2.

i.e. $\varepsilon(u) = y(u) - y_s(u) = 0$ QED

Remarks

R3-1. The separability notion introduced here is in some sense, a definition of the linearization exactness. The output of a nonlinearity, which is not strictly separable from order m on, gives harmonic components other than the m inputs. For example, the sinusoidal input case is not separable for the first order (except the linearity); if $f(x) = x^3$ and $x = \sin \omega t$ then $f(x) = (1/4)(3 \sin \omega t - \sin 3\omega t)$, the output contains a component $\sin 3\omega t$.

R3-2. For an odd nonlinearity the Walsh describing method is not strictly separable, in general, from the order 3 on. For example, if $f(x) = x^3$ and $x = \sum_{i=1}^3 a_i \text{wal}(k_i, u)$ then $f(x)$ contains a distortion term $6a_1a_2a_3 \text{wal}(k_1+k_2+k_3, u)$. We will consider the case in more detail in section 4.

R3-3. The first part of the theorem 3 may be important for the study of unbalanced nonlinearity cases.

On the other hand the linearization separability of Walsh functions input suggests a synthesis of a group of nonlinearities whose parallel, as well as, series combination has the identical input-output characteristics.

PARALLEL COMBINATION

Consider q nonlinearities f_1, f_2, \dots, f_q in parallel such that the output $y(x)$ is

$$y(x) = \sum_{j=1}^q f_j(x).$$

It then follows that the WIDF for the composite nonlinearity is given by:

$$N(a) = \sum_{j=1}^q N_j(a) \text{ for } x = a \text{wal}(k, u), \quad (42)$$

and

$$\begin{aligned} N_{a_1}(a_1, a_2) &= \sum_{j=1}^q N_{j, a_1}(a_1, a_2) \\ \text{for } x &= \sum_{i=1}^q a_i \text{wal}(k_i, u) \end{aligned} \quad (43)$$

where $N_{j, a_1}(a_1, a_2)$ is the Walsh DF for the j -th nonlinearity corresponding to the input component $a_1 \text{wal}(k_1, u)$.

SERIES COMBINATION [4]

For simplicity, consider 2 nonlinearities f_1, f_2 in series such that the composite output is

$$y(x) = f_2[f_1(x)]$$

If the input $x(u)$ is $x(u) = a \text{wal}(k, u)$ then the output $z(u)$ of the 1st nonlinearity f_1 is equal to

$$z(u) = f_1[x(u)] = aN_1(a) \text{wal}(k, u). \quad (45)$$

$$\text{Let } z(u) = a_z \text{wal}(k, u). \quad (46)$$

Then, it follows that

$$\begin{aligned} y(u) &= f_2[z(u)] \\ &= a_z N_2(a_z) \text{wal}(k, u). \end{aligned} \quad (47)$$

From (47), (46) and (45) the WIDF for the composite nonlinearity is given by

$$\begin{aligned} N(a) &= a_z N_2(a_z) / a \\ &= aN_1(a)N_2(aN_1(a)) / a \\ N(a) &= N_1(a)N_2(aN_1(a)). \end{aligned} \quad (48)$$

If the input is now $x(u) = a_1 \text{wal}(k_1, u) + a_2 \text{wal}(k_2, u)$ then in a similar manner, we get

$$\begin{aligned} N_{a_1}(a_1, a_2) &= N_{1, a_1}(a_1, a_2)N_{2, a_1}(aN_{1, a_1}(a_1, a_2), \\ &\quad a_2N_{1, a_2}^2(a_1, a_2)) \end{aligned} \quad (49)$$

for $i = 1, 2$.

This results (48) and (49) could be generalized for q nonlinearities in series.

Using (42, 48) or (43, 49) we would have a composite parallel-series combination.

TABLE 1

wal(k,u)	$2a_0N_0(a_0, a_1)$	$2a_1N_1(a_0, a_1)$	wal(k,u)	$y_s(u)$	$y(u)$
1	$f(a_0+a_1)+f(a_0-a_1)$	$f(a_0+a_1)-f(a_0-a_1)$		$f(a_0+a_1)$	$f(a_0+a_1)$
-1	$f(a_0+a_1)+f(a_0-a_1)$	$-f(a_0+a_1)+f(a_0-a_1)$		$f(a_0-a_1)$	$f(a_0-a_1)$

TABLE 2

wal(k ₁ ,u)	wal(k ₂ ,u)	$2a_1N_1wal(k_1,u)$	$2a_2N_2wal(k_2,u)$	y_s	y
1	1	$f(a_1+a_2)+f(a_1-a_2)$	$f(a_1+a_2)-f(a_1-a_2)$	$f(a_1+a_2)$	$f(a_1+a_2)$
1	-1	$f(a_1+a_2)+f(a_1-a_2)$	$-f(a_1+a_2)+f(a_1-a_2)$	$f(a_1-a_2)$	$f(a_1-a_2)$
-1	1	$-f(a_1+a_2)-f(a_1-a_2)$	$f(a_1+a_2)-f(a_1-a_2)$	$-f(a_1-a_2)$	$f(-a_1+a_2)$
-1	-1	$-f(a_1+a_2)-f(a_1-a_2)$	$-f(a_1+a_2)+f(a_1-a_2)$	$-f(a_1+a_2)$	$f(-a_1-a_2)$

DESCRIBING FUNCTION FOR FINITE-SUM ARBITRARY INPUT (AIDF)

We know that $\{wal(k,u); k, u = 0, 1, \dots, 2^n-1\}$ is a complete orthogonal function set [3], and, in general, an arbitrary function can be expressed as a sum of an infinite number of Walsh functions [2].

1. Generality

Consider now the case where the input $x(u)$ is a finite sum:

$$x(u) = \sum_{k=0}^{2^n-1} a_k wal(k,u) \quad (50)$$

where all a_k are not necessarily different from zero.

The reason for choosing the expression (50) is because it minimized the mean squared error $|x_{in}(u) - x(u)|^2$ when a_k is the finite Walsh Fourier transformation of $x_{in}(u)$. x_{in} is an arbitrary input and $P=2^n$ is fixed. This fact is well known and may be illustrated by the figure 2. Of course, the bigger P , the smaller the mean-squared error.

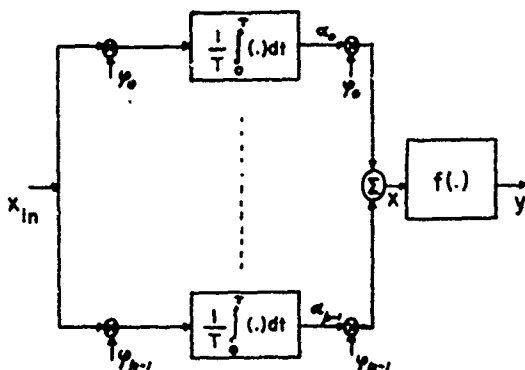


Fig. 2 - Finite Walsh-Fourier Transform Block

Now if the input to the nonlinearity is $x(t)$ (50) then the approximate output is $y_s(t)$, and by means of (11), it may be written as

$$y_s(u) = \sum_{k=0}^{P-1} a_k N_k(a_0, \dots, a_{P-1}) wal(k,u). \quad (51)$$

The problem is still finding the set $N_k(a_0, \dots, a_{P-1})$ or, precisely, the set $a_k N_k(a_0, \dots, a_{P-1})$, because if some a_k in (50) is equal to zero, writing N_k alone is then meaningless.

We denote that the number m of zero mean inputs is equal to $m = 2^n - 1 > n$. So, we cannot apply the results of (29) and (30). In order to avoid this difficulty, consider a more general formula:

$$a_k N_k \triangleq a_k N_k(a_0, \dots, a_{P-1}) \quad (52)$$

$$f(u) \triangleq f\left(\sum_{k=0}^{P-1} a_k wal(k,u)\right) \quad (53)$$

Thus, from (15) and (50) we have

$$a_k N_k = (1/P) \sum_{n=0}^{P-1} f(u) wal(k,u) = \frac{1}{P} [wal(k,0) wal(k,1) \dots wal(k,P-1)] \begin{bmatrix} f(0) \\ f(1) \\ \dots \\ f(P-1) \end{bmatrix}$$

or

$$[aN] = \begin{bmatrix} a_0 N_0 \\ a_1 N_1 \\ \dots \\ a_{P-1} N_{P-1} \end{bmatrix} = \frac{1}{P} [W_P] \begin{bmatrix} f(0) \\ f(1) \\ \dots \\ f(P-1) \end{bmatrix}, \quad (54)$$

where $[W_P]$ is the $2^n \times 2^n$ Walsh matrix.

But, in matrix notation, (53) means

$$f(u) = f[wal(0,u) \dots wal(P-1,u)] \begin{bmatrix} a_0 \\ a_1 \\ \dots \\ a_{P-1} \end{bmatrix} \quad (55)$$

By the symmetry of Walsh matrix, so let us define

$$[f([W_p][a])] = \begin{bmatrix} f(0) \\ f(1) \\ \dots \\ f(P-1) \end{bmatrix} \quad (56)$$

$$\text{where } [a] = [a_0 a_1 \dots a_{P-1}]^T. \quad (57)$$

Thus (54) becomes

$$[aN(a_0, \dots, a_{P-1})] = (1/P)[W_p][f([W_p][a])]. \quad (58)$$

2. Linearization Separability

Theorem 4: Closeness of Nonlinearity Output Components

*Let the input be (50) a linear combination of a complete orthogonal set $\{wal(k,u); k,u = 0, 1, \dots, P-1\}$ where all a_k are not necessarily different from zero. The output of any zero-memory nonlinearity $f(x)$ may be written as:

$$y[x(u)] = y_s(u) = \sum_{k=0}^{P-1} (a_k N_k) wal(k,u) \quad (59)$$

where the set $\{(a_k N_k)\}$ is given by (54) or (58).

*Let the input be (50). If $a_k = 0$ for k such that the greatest integer part $[(k+1)/2]$ is even, and if $f(x)$ is odd, then

$$y[x(u)] = y_s(u) = \sum_{k=0}^{P-1} (a_k N_k) wal(k,u) \quad (60)$$

for $(k+1)/2$ odd

Equations (59) and (60) each represent P expressions $[u=0, 1, \dots, P-1]$.

Proof: *Let the column matrix $[y_s(u)]$ be

$$[y_s(u)] = [y_s(0) \dots y_s(P-1)]^T. \quad (61)$$

From (51) and (54), it then follows that

$$\begin{aligned} [y_s(u)] &= [W_p][aN] \\ &= (1/P)[W_p][f([W_p][a])] \\ &= [f(u)]. \end{aligned}$$

*The second part of this theorem will be shown in two steps:

$$(a) \text{ Express } \sum_{k=0}^{P-1} a_k wal(k,u) \quad (62)$$

where $(k+1)/2$ is odd

in a linear combination of Hadamard (Walsh-Paley) functions $h_n(j,u)$ such that

$$x(u) = \sum_{j=P/2}^{P-1} b_j h_n(j,u) \quad (63)$$

where [3]

$$h_n c_j(u) \hat{=} (1-1)^{\sum_{r=0}^{n-1} j_r u_r} \quad \text{and} \quad (64)$$

$$j = \sum_{r=0}^{n-1} j_r 2^r, \quad u = \sum_{r=0}^{n-1} u_r 2^r; \quad j_r, u_r \in \{0,1\} \quad (65)$$

(b) From (63) apply the result of the first part of the theorem.

Let us define the following bijection Γ (one-to-one map)

$$\Gamma: j_0 \dots j_{n-1} \text{ such that } wal(k,u) = h_n(j,u) \quad (66)$$

In comparing (17) and (64) we see that

$$\Gamma: j_0 = k_{r-1}, \quad j_r = k_{n-r} \oplus k_{n-r-1} \quad (67)$$

and the inverse $\Gamma^{-1}: k_r = \sum_{j=0}^{n-r-1} j_s$

The mapping Γ permutes the set $\{a_k; k=0, \dots, P-1\}$ into $\{b_j; j=0, \dots, P-1\}$. In order to show (63), from (56), it is thus sufficient to show $j=P/2, \dots, P-1$ or $j_{n-1}=1$.

The condition $[\frac{1}{2}(k+1)] = 2p+i$, $(p: \text{positive integer})$ (68)

may be written as follows:

$$k = \begin{cases} 4r+1 & \text{for } k \text{ odd} \\ 4p+2 & \text{for } k \text{ even} \end{cases} \quad (69)$$

But $k = 0, 1, 2, \dots, 2^n-1$, so (69) gives

$$p = 0, 1, 2, \dots, 2^{n-1}-1, \text{ i.e.,}$$

$$p = \sum_{r=0}^{n-3} p_r 2^r = \sum_{r=0}^{n-1} p_r 2^r \text{ with } p_{n-1} = p_{n-2} = 0 \quad (70)$$

Let us write

$$p = (0, 0, p_{n-3}, p_{n-4}, \dots, p_0). \quad (71)$$

From (69) it then follows that

$$k = \begin{cases} (p_{n-3}, p_{n-4}, \dots, p_0, 0, 1) & \text{for } k \text{ odd} \\ (p_{n-3}, p_{n-4}, \dots, p_0, 1, 0) & \text{for } k \text{ even} \end{cases} \quad (72)$$

From (67), (72) implies

$$j_{n-1} = k_0 + k_1 = 1, \text{ i.e.,}$$

$$j = P/2, (P/2) + 1, \dots, P-1.$$

(b) In matrix notation, (63) means

$$\begin{bmatrix} x(0) \\ \vdots \\ x(P/2-1) \\ x(P/2) \\ \vdots \\ x(P-1) \end{bmatrix} = \begin{bmatrix} H_{P/2} & H_{P/2} \\ H_{P/2} & -H_{P/2} \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ b_{P/2} \\ \vdots \\ b_{P-1} \end{bmatrix} \quad (73)$$

where $H_{P/2}$ is the $(2^{n-1} \times 2^{n-1})$ Hadamard matrix $[h_{n-1}(j,u)]$. Thus, it is clear that for $u=0, 1, \dots, P/2-1$

$$x(u) = \sum_{j=P/2}^{P-1} b_j h_n(j,u) = \sum_{j=0}^{P/2-1} b_{P/2+j} h_{n-1}(j,u)$$

and for $u=P/2, \dots, P-1$

$$x(u) = - \sum_{j=0}^{P/2-1} b_{P/2+j} h_{n-1}(j,u)$$

i.e.,

$$x(u + P/2) = -x(u), \text{ for } u = 0, 1, \dots, P/2-1 \quad (74)$$

Moreover, the nonlinearity $f(x)$ is supposed odd. It then follows that

$$f[x(u+P/2)] = -f[x(u)]; \quad u=0, 1, \dots, P/2-1 \quad (75)$$

The approximate output

$$y_s(u) = \sum_{k=0}^{P-1} (a_k N_k) \text{wal}(k, u),$$

$$\left\{ \frac{k+1}{2} \right\} \text{ odd}$$

with the mapping $\Gamma(k) = j$, becomes

$$y_s(u) = \sum_{j=P/2}^{P-1} (b_j M_j) h_n(j, u) \quad (76)$$

where $b_j M_j = a_k N_k$

$$= (1/P) \sum_{u=0}^{P-1} f[x(u)] \text{wal}(k, u); \text{ from (66)}$$

$$= (1/P) \sum_{u=0}^{P-1} f[x(u)] h_n(j, u); \text{ from (75)}$$

$$b_j M_j = (2/P) \sum_{u=0}^{P/2-1} f[x(u)] h_{n-1}(j, u)$$

for $j = P/2, \dots, P-1$. (77)

From (76) it is not difficult to see that $y_s(u)$ is also rotational symmetric:

$$y_s(u + P/2) = -y_s(u); u = 0, 1, \dots, P/2-1 \quad (78)$$

Consider then u from 0 to $P/2-1$:

In matrix notation

$$[y_s(u)] = [H_{P/2}] [bM] \quad (79)$$

where $[bM]$, from (77), is equal to

$$[bM] = (2/P) [H_{P/2}] [f(u)] \quad (80)$$

with $f(u) \triangleq f[x(u)]$.

Replacing (80) into (79) results in:

$$[y_s(u)] = [f(u)]. \quad \text{QED}$$

Remarks

R4-1. The theorem 4 explains the counter example in the previous section (R3-2). By taking $k_1=1$, $k_2=2$, $k_3=3$ the distortion term is $6a_1 a_2 a_3 \text{wal}(0, u)$.

If all a_k is different from zero, (59) means that the output of any zero-memory non-linearity is strictly P-separable. And particularly the output of odd nonlinearity is said also strictly separable for odd sequency components, if the $a_k \neq 0$ for k such that $\{(k+1)/2\}$ is odd.

R4-2. The theorem 4 may yield a better means for identification and/or synthesis of zero-memory nonlinearity elements.

For example, consider 2 nonlinearities f_1, f_2 in series (44). It is easy to generalize (49) for an arbitrary input of the form (50):

The composite output amplitude of the k^{th} Walsh component will be

$$a_k N_k(a_0, \dots, a_{P-1}) = a_k N_{1,k}(a_0, \dots, a_{P-1})$$

$$N_{2,k}(a_0 N_{1,0}, \dots, a_{P-1} N_{1,P-1}) \quad (81)$$

CONCLUSION

An attempt has been made to study nonlinearities using a Walsh function input. It has been established that the resulting describing function is always a real function, and that it is separable up to order two. This linearisation separability permits the synthesis of a group of nonlinearities (series or parallel combination) in an easy fashion by means of a relatively simple closed form expression. An arbitrary input can be handled in a similar manner, but the manipulations are more complex.

REFERENCES

- [1] J.C. HSU & A.U. MEYER, "Modern Control Principles and Applications", McGraw Hill, 1968, pp. 209-212.
- [2] J.L. WALSH, "A Closed Set of Normal Orthogonal Functions", Am. Jour. of Math., vol. 45, pp. 5-24, 1923.
- [3] J.E. GIBBS, "Sine Waves and Walsh Waves in Physics", presented at the 1970 Workshop on Applications of Walsh Functions, Naval Research Laboratory, Washington, D.C., March-April 1970.
- [4] A. GELB & W. VANDER VELDE, "Multiple-Input Describing Functions and Nonlinear System Design", McGraw Hill, 1968.

APPLICATIONS OF WALSH FUNCTIONS TO NUMBER THEORY

R. Binkin and E. L. Hall
University of Missouri - Columbia

Abstract

Results are presented in this paper of digital pictorial representations of various number sequences and their respective Walsh transforms. Fibonacci, prime number, and simple geometric sequences and point patterns are generated as square binary arrays. The array elements are set equal to 1 if the sum of its coordinates is equal to a number in the particular sequence. All other array elements are set equal to 0. The Walsh transforms are then computed. These pictorial representations reveal certain patterns which are not obvious without the pictorial representations and the Walsh transforms of the arrays. These results indicate a method in which digital image processing methods and the Walsh transforms may be applied to describe and develop patterns related to number theory.

Introduction

The use of pictures to study number theory is not new. In fact over 2000 years ago Pythagoras used pictures to develop certain properties. For example, starting with a single dot, how many dots can be added to form a triangle? The answer shown in Fig. 1 leads to a simple development of the formula for the sum of the first n consecutive integers.

$$\sum_{i=0}^n i = \frac{n(n+1)}{2}$$

Similarly, starting with a single dot, how many dots can be added to form a square? This answer is shown in Fig. 2 and again leads to a formula which states

$$\sum_{i=1}^n (2i-1) = n^2$$

that the sum of the first n consecutive odd integers is a perfect square.

Manfred Schroeder [1] expanded on this concept of using pictures to study number theory by using computer image processing techniques to display both graphic representations of number functions and their Fourier transform in order to attempt to reveal interesting geometric patterns.

In this paper we will further Schroeder's work by graphically displaying various number series but instead of computing the Fourier transform, we compute the sequency ordered Walsh transform since it is better suited for

the binary functions that will be generated. The series that were investigated were simple point patterns, geometrics series, Fibonacci series, and prime number series. Each one was redefined into a two-dimensional function and the Walsh transforms were then generated.

The Sequency Ordered Walsh Function

The Walsh transform [2] is being used increasingly in image processing [3].

The forward Walsh transform is given by [4]

$$F(u,v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) a(x,y,u,v)$$

where $a(x,y,u,v)$ is the forward transformation kernel. For our use, we will express the two dimensional transform in matrix notation. For a transform kernel that is separable symmetric let

$[f]$ = image matrix
 $[F]$ = transformed image matrix
 $[A]$ = transform matrix

Then by matrix multiplication

$$[F] = [A] [f] [A]$$

$[A]$ is an $N \times N$ sequency ordered Walsh matrix [5]. An example of a 4×4 Walsh matrix is

$$W = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

and has the following properties:

1. The matrices are square with dimension $2^N \times 2^N$, $N = 1, 2, 3, \dots$
2. The dot product of any two rows is zero.
3. $W^{-1} = W^T$, since the transform is unitary.
4. The rows are ordered by increasing the number of sign changes.

Prime Numbers

A 4×4 array can represent a prime number series if we define the array $f(x,y)$ as

$$f(x,y) = \begin{cases} 1 & \text{if } N=x+4y \text{ and is a prime number} \\ 0 & \text{if } N=x+4y \text{ and is a non-prime number} \end{cases}$$

Fig. 3 is a pictorial representation of this array. Fig. 4 shows how the sequency ordered Walsh is calculated. Fig. 5 and 6 show digitally processed images of both the original image and its Walsh transform. In order to implement the ordered Walsh transform with differently sized arrays, the $n \times n$ array will expand out to a 128×128 array by addition of zeros (see Fig. 7). This 128×128 expanded matrix was operated on by a 128×128 ordered Walsh matrix to generate the Walsh transforms.

In prime number arrays of size $(N \times N)$, when N equals non-prime numbers, vertical bands tend to stand out as in the cases where $N = 30, 40$ and 42 , shown in Figs. 8-10. When N equals a prime number, diagonal bands become more apparent as with $N = 31, 43$, shown in Figs. 11 and 12.

The Walsh transforms of these $(N \times N)$ prime number arrays fall into one of two categories, either high contrast and low contrasted array elements; no pattern is apparent between the category of the Walsh transform and the value of N , but many of these transform arrays have some interesting patterns that can most effectively be observed when displayed graphically as in Figs. 13-20.

Points Patterns and Line Patterns

The sequency ordered Walsh transform of point and line patterns with definite geometric patterns also contain definite geometric patterns. To illustrate this, 128×128 arrays were generated where

$$f(x,y) = \begin{cases} 1 & \text{when } x = iN \\ & y = jM \\ & \text{where } i, j = 1, 2, \dots \\ 0 & \text{for all others} \end{cases}$$

For each of these arrays, the Walsh transforms were calculated.

It becomes obvious that the Walsh transform of point patterns with any x and y frequencies can be defined in terms of the superposition of the Walsh transform of vertical lines of N frequency and horizontal lines of M frequency.

The Walsh transform of a set of lines is an array of all zeros except in the first row or column perpendicular to the set of lines. Examples of sets of line patterns with $M = 1$ and $N = 1-5$ and their Walsh transforms are shown in Figs. 21-30. By rotating the set of lines by 90° the Walsh transform is rotated also by 90° as illustrated in Figs. 31-32. Notice the simple patterns of the Walsh

transform when N equals a binary number as compared to Walsh transforms when N equals a non-binary number. If the array is a point pattern, $N > 1$ and $M > 1$, the Walsh transform appears to be equal $f(u,v) = g(u,1) \times h(1,v)$, where $g(x,y)$ is the Walsh transform of a set of vertical lines and $h(x,y)$ is the Walsh transform of horizontal lines. Figs. 33 and 34 illustrate this with a point pattern array where $N = 4$ and $M = 3$ and its Walsh function.

Power Series

Another interesting set of series to study with image analysis is the power series. The two dimensional array can be defined as

$$f(x,y) = \begin{cases} 1 & \text{when } x + (128yy) = P^I \text{ where } P \\ & \text{equals an arbitrary constant and} \\ & I \text{ equals any integer.} \\ 0 & \text{for all others} \end{cases}$$

In these arrays there are no evident patterns except in the cases where 128 is a multiple of P . In this case, the first row and the last column of each array are equal and all other elements of the array are equal to zero.

The Walsh transform of the power series have similar characteristics between them. They all tend to have vertical bands of differing textures. When I is odd, there is definite oscillating in the texture. When I equals an even number, the texture tends to be more even. Figs. 35-38 are illustrations of array representations of the power series for $I = 2$ and 3 and these Walsh transforms. As P becomes larger, the vertical bands become more obscure.

Fibonacci Series

The Fibonacci series is generated by the formula $F(N) = F(N-1) + F(N-2)$ where $F(1) = 1$ and $F(2) = 2$. This can be put into a two dimensional array by setting $I = X + 128y$ and

$$f(x,y) = \begin{cases} 1, & \text{when } I \text{ is a Fibonacci number} \\ 0, & \text{otherwise} \end{cases}$$

This array is pictured in Fig. 39 and its Walsh transform is pictured in Fig. 40.

Conclusion

In this paper we have considered various number series. Each series was redefined into a two dimension array and the ordered Walsh transform was computed. The hope was to find geometric patterns in either the two dimensional array representation or their Walsh transforms. In the case of point patterns definite patterns were found. In the other cases definite patterns were obscured yet some general patterns were still discovered. Further work in this area could possibly prove to yield some interesting discoveries in number theory and if nothing else would improve one's intuitive feel for the Walsh transform and its applications with respect to pattern recognition.

References

- [1]. Schroeder, M.R., "Images from Computers," IEEE Spectrum, March, 1969.
- [2]. Walsh, J.L., "Remarks on the History of Walsh Functions," Proc. of the Walsh Function Symposium, Washington, D.C., April, 1970.
- [3]. Pratt, W.K., "Linear and Nonlinear Filtering in the Walsh Domain," Univ. of Southern California, Dept. of Elect. Eng., Los Angeles, California.
- [4]. Andrews, H.C. and Pratt, W.K., "Digital Image Transform Processing," Applications of Walsh Functions Symposium and Workshop, Washington, D.C., April, 1970.
- [5]. Walsh, D.M., "Walsh Domain Filter Techniques," Master's Thesis in Engineering, Univ. of South Florida, March, 1970.

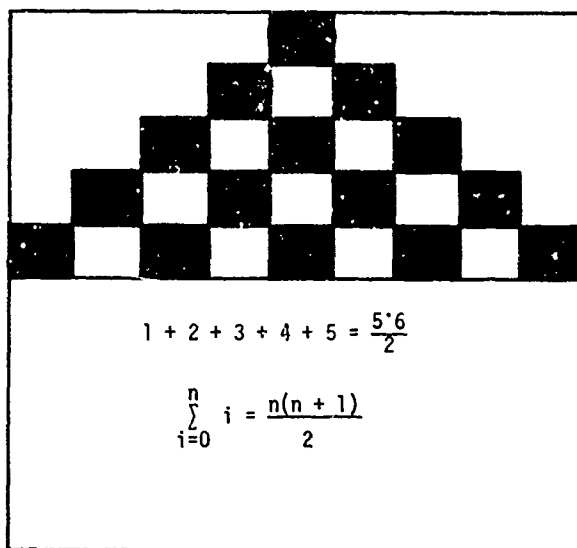


Fig. 1. Triangular numbers

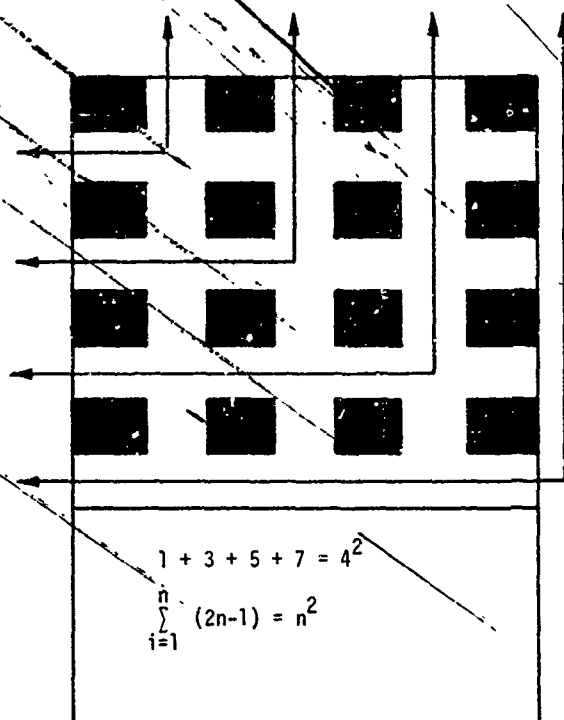


Fig. 2. Rectangular numbers

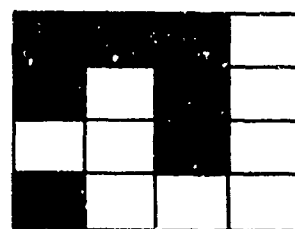


Fig. 3. Prime numbers indicated on a 4 x 4 array

$$F = W P W$$

$$F = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & -1 \end{bmatrix}$$

$$F = \begin{bmatrix} 7 & 1 & -1 & 5 \\ 3 & 1 & -1 & 1 \\ 1 & 3 & 1 & -1 \\ 1 & -1 & -3 & -1 \end{bmatrix}$$

Fig. 4. Ordered Walsh transform computation for 4 x 4 array

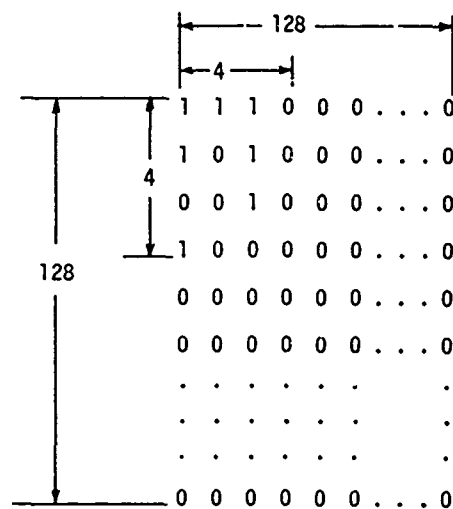


Fig. 7. Expanded 4 x 4 array

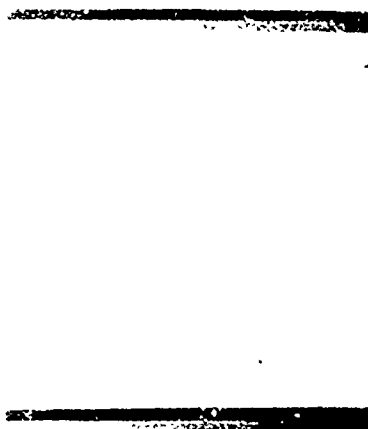


Fig. 5. 4 x 4 array of prime numbers



Fig. 8. 30 x 30 array of prime numbers

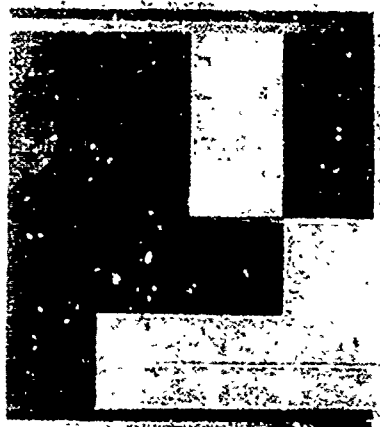


Fig. 6. Walsh transform

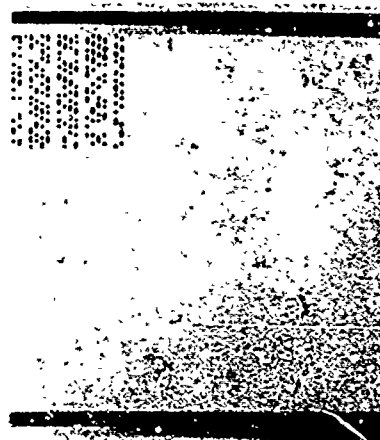


Fig. 9. 40 x 40 array of prime numbers

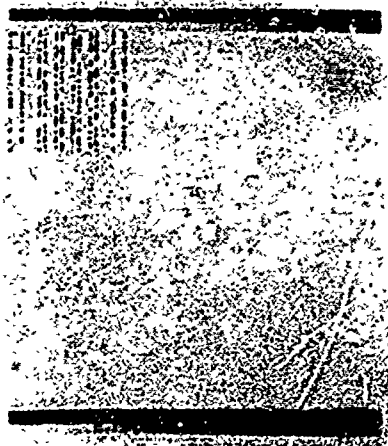


Fig. 10. 42 x 42 array of prime numbers



Fig. 13. 13 x 13 array of prime numbers

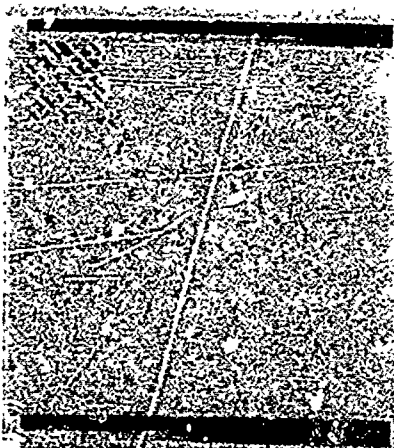


Fig. 11. 31 x 31 array of prime numbers

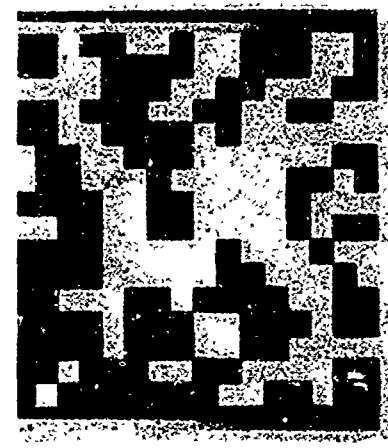


Fig. 14. Walsh transform of Fig. 13

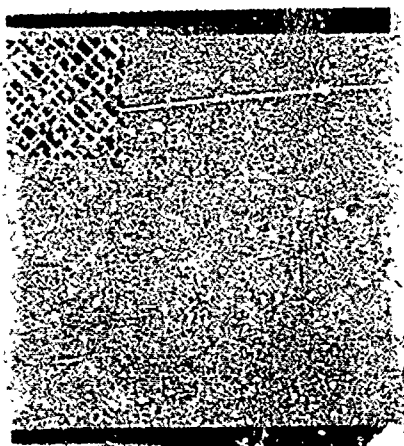


Fig. 12. 43 x 43 array of prime numbers

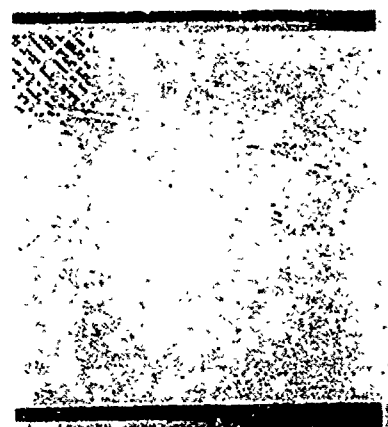


Fig. 15. 31 x 31 array of prime numbers

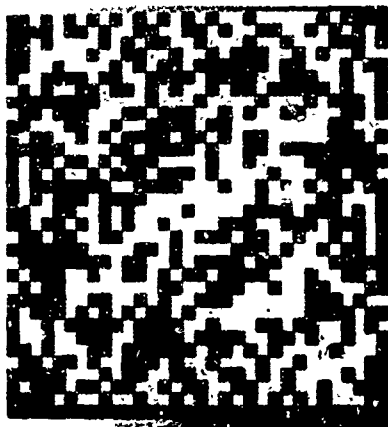


Fig. 16. Walsh transform of Fig. 15

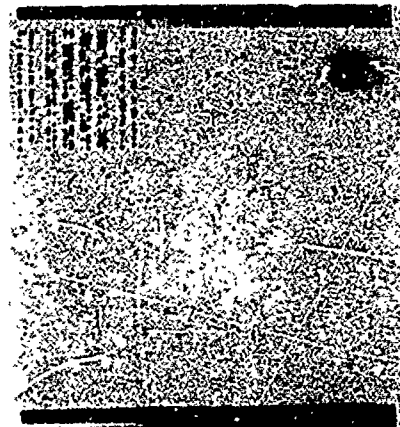


Fig. 19. 42 x 42 array of prime numbers



Fig. 17. 36 x 36 array of prime numbers

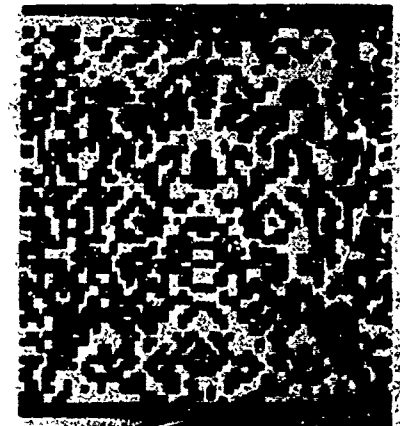


Fig. 20. Walsh transform of Fig. 19

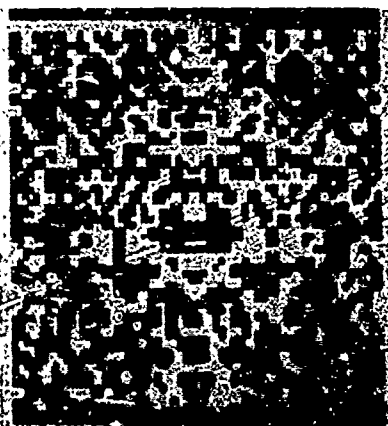


Fig. 18. Walsh transform of Fig. 17



Fig. 21. Point pattern $N = 1$ and $M = 1$

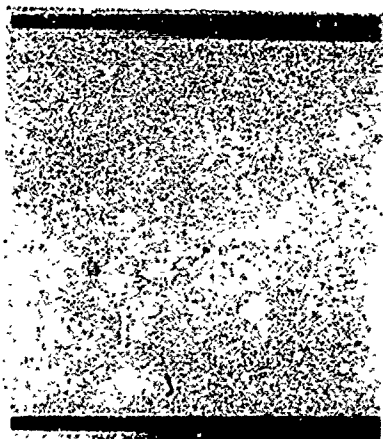


Fig. 22. Walsh transform of Fig. 21

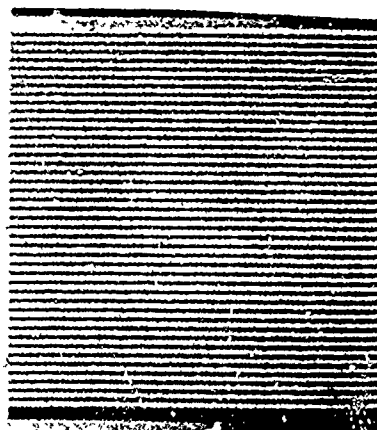


Fig. 25. Point pattern $h = 3$ and $M = 1$

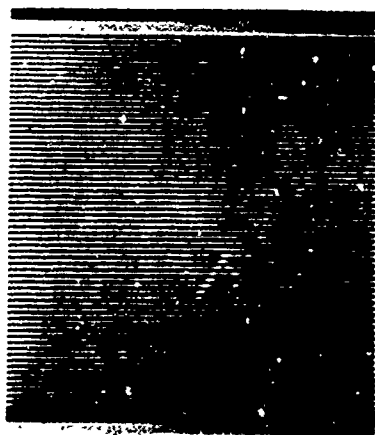


Fig. 23. Point pattern $N = 2$ and $M = 1$

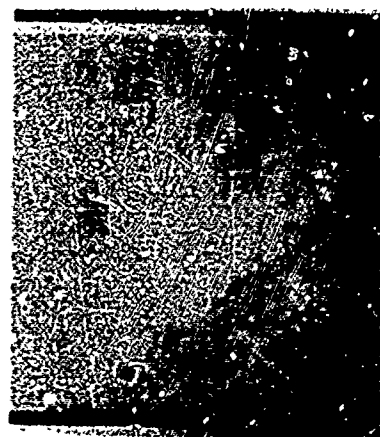


Fig. 26. Walsh transform of Fig. 25



Fig. 24. Walsh transform of Fig. 23

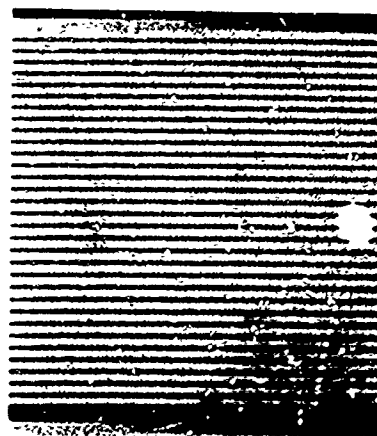


Fig. 27. Point pattern $N = 4$ and $M = 1$



Fig. 28. Walsh transform of Fig. 27

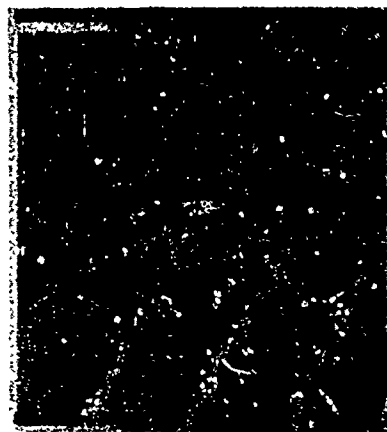


Fig. 31. Point pattern $N = 1, M = 2$

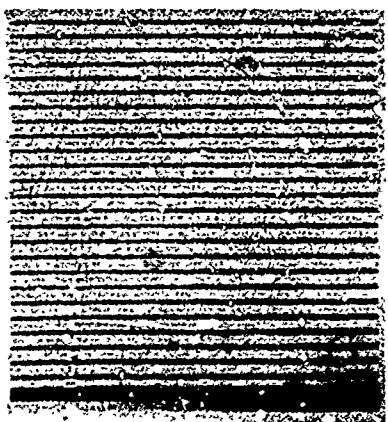


Fig. 29. Point pattern $N = 5$ and $M = 1$



Fig. 32. Walsh transform of Fig. 31



Fig. 30. Walsh transform of Fig. 29

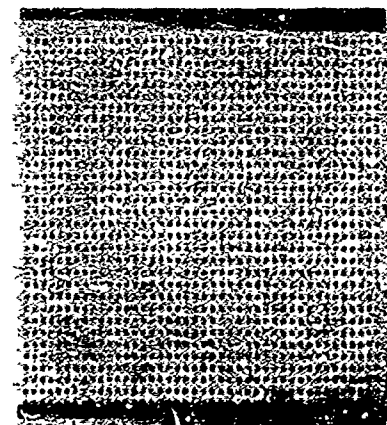


Fig. 33. Point pattern $N = 4$ and $M = 3$

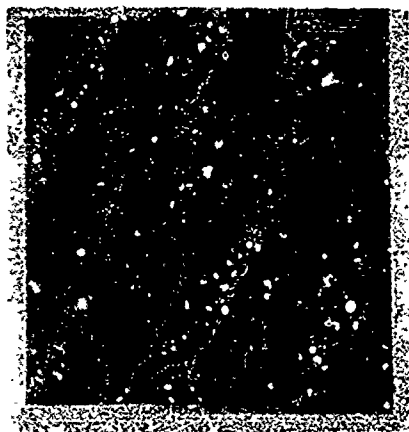


Fig. 34. Walsh transform of Fig. 33



Fig. 37. Power series, $I = 3$

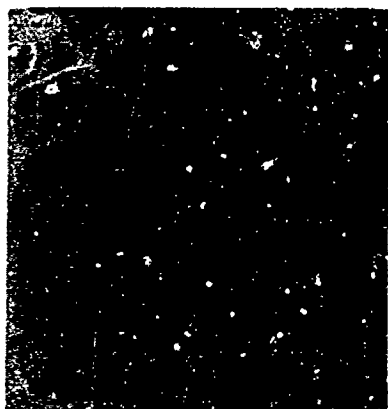


Fig. 35. Power series, $I = 2$

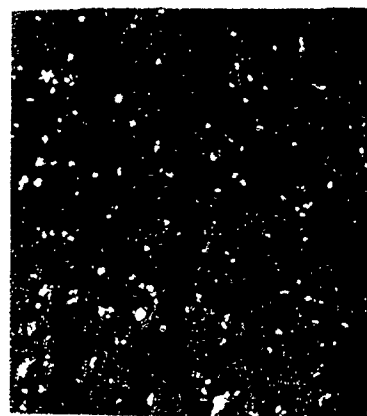


Fig. 38. Walsh transform of Fig. 37

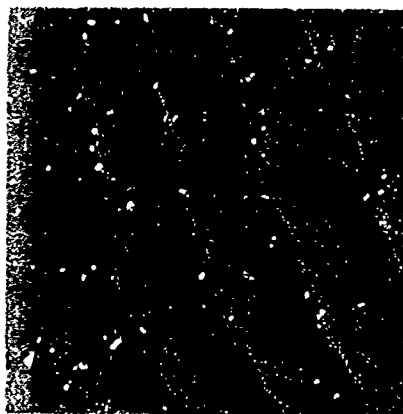


Fig. 36. Walsh transform of Fig. 35

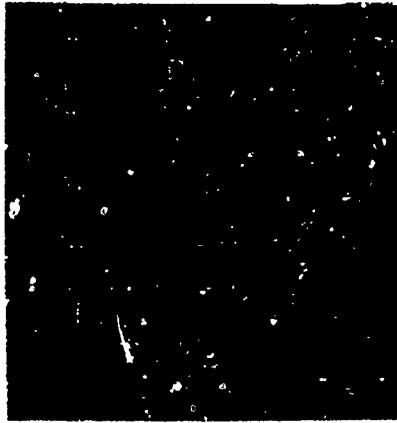


Fig. 39. Fibonacci series array

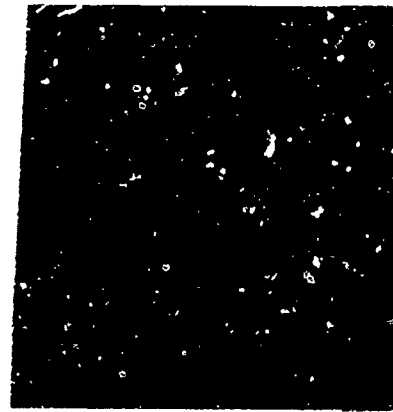


Fig. 40. Walsh transform of Fig. 39

ON CLASSIFICATION OF BINARY SEQUENCES

Subhash C. Kak
Department of Electrical Engineering
Indian Institute of Technology
New Delhi-29, India

Reproduced from
best available copy.

Abstract

Measures to characterize randomness in finite binary sequences have been proposed by various authors. Amongst these are the one's obtained through the computational complexity approach and the integral transform approach. The latter measure which is given by the ratio of the number of non-zero Walsh coefficients of the sequence to its length, is unambiguous and readily calculated. Further this procedure seems to give results similar to those obtained through the computational complexity approach. This paper is an attempt at explaining the efficacy of the Walsh transform measure.

The result on the number of long random binary sequences is further analyzed. The measure is also generalized to cover other transforms. This is illustrated using discrete Legendre transformation. It is shown that for this the procedure becomes cumbersome and the results less amenable to meaningful interpretation.

Introduction

Recently there has been a considerable interest in the computational - complexity approach to the definition of a finite random sequence pioneered by Solomonoff [1], Kolmogorov [2], [3] and Chaitin [4], [5]. Instead of testing a given sequence for regularity through sub-sequence generation, the approach involves directly asking if the sequence is irregular or complex. The notion of complexity is taken to correspond to the length of the shortest program capable of generating the sequence on a certain asymptotically optimal machine. On the basis of this definition those elements of a finite population whose complexity is maximal are called random. It has been shown that most sequences in a population are random. Per Martin-Löf [6] has also shown that the above definition leads to all conceivable statistical properties of randomness. For infinite random binary sequences, the non-random sequences have been shown to form a maximal constructive null set. Fine [7] has elaborated these conclusions for finite

sequences. Apparent convergence of the relative frequency is shown to occur, because of, and not in spite of, the high irregularity (or randomness) of the data sequence. Further the stability of relative frequency is due to the approach to data and not according to some law of nature. Consequently it appears that apparent stability in frequency cannot be used for extrapolation of future data.

Another approach which has been employed in the representation of finite binary sequences uses orthonormal functions [8], [9]. This very straightforward procedure involves the representation of the given sequence in a weighted sum of Walsh functions. The number of such terms required per sequence digit is then a measure of the randomness of the sequence. This approach seems to yield results much similar to the computation-complexity approach. The Walsh function expansion procedure also leads to the result that most sequences are random and therefore seems to suggest the same conclusions as listed in the preceding paragraph.

Investigating the apparent similarities of the two approaches seems to be a worthwhile problem. The computation complexity approach has its roots in the work of Church [10] who used the concept of effective computability to define infinite random sequences. However this definition turns out to be different from that obtained in the former, being in fact, only a clarification of the notion of the kollektiv of von Mises [11]. The kollektiv is an infinitely long random sequence in which limits of relative frequencies exist and there is no scheme of predicting outcome x_n of trial n given knowledge of x_1, x_2, \dots, x_{n-1} , that yields infinite subsequences having relative frequencies that differ from those of the original sequence. The idea of subsequence generation was also taken up by Karl Popper [12] who however felt that a more satisfactory approach to the definition of an infinite random sequence should be a generalization of a random finite sequence. In a loose fashion, the authors approach [3] of representing the given sequence through a Walsh function expansion draws from

the notion of sub-sequence generation. It also draws from the computation complexity approach in being an attempt at compressing data (or dimension contraction).

An important question that arises now is why should the Walsh transformation be chosen for dimension contraction in preference to other transforms. The Walsh functions correspond to only one out of a great many possible vector bases for 2^k -dimensional space and each basis would yield a different measure of randomness as a generalization of this measure. Therefore a rationale must be found for the efficacy of Walsh functions, or this measure generalized in some manner to include all bases. In our view the former appears more likely. In this paper an attempt has been made to justify this belief. An instance, where the measure of randomness using Walsh transforms is extended to include discrete Legendre transforms is also studied. It is shown how the analysis becomes involved for this case.

Measure Using Walsh Transform

Let $w_k(t)$ represent the k th discrete Walsh function of t , where k and t are non-negative integers smaller than $N = 2^n$. Also let

$$(k)_{\text{decimal}} = (k_{n-1} k_{n-2} \dots k_0)_{\text{binary}}$$

and

$$(t)_{\text{decimal}} = (t_{n-1} t_{n-2} \dots t_0)_{\text{binary}}$$

then the Walsh function set is defined through

$$w_k(t) = (-1)^{\sum_{i=0}^{n-1} g_i(k) t_i} \quad (1)$$

For natural ordered Walsh functions

$$g_1(k) = k_1 \quad (2)$$

while for frequency ordered functions, $g_1(k)$'s are defined through the following relations:

$$g_0(k) = k_{n-1}$$

$$g_1(k) = k_{n-1} + k_{n-2}$$

$$g_2(k) = k_{n-2} + k_{n-3}$$

$$\dots \dots \dots$$

$$g_{n-1}(k) = k_1 + k_0 \quad (3)$$

In what follows the Walsh functions may be used either in the natural ordering or the frequency ordering. Therefore we would not differentiate between them and refer to either of them as $w_k(t)$.

For $s(t)$, an arbitrary bounded function of the discrete parameter t , ($t = 0, 1, \dots, 2^n - 1$), the Walsh transform $s^*(k)$ is defined by

$$s^*(k) = N^{-1} \sum_t w_k(t) s(t) \quad (4)$$

The inverse transformation is symmetrical:

$$s(t) = N^{-1} \sum_k w_k(t) s^*(k) \quad (5)$$

Given a binary sequence $s(t)$, we first find its Walsh transform $s^*(k)$ and then obtain another sequence $u_s(k)$ through the following procedure:

$$u_s(k) = \begin{cases} 1 & \text{if } s^*(k) \neq 0 \\ 0 & \text{if } s^*(k) = 0 \end{cases} \quad (6)$$

Then the measure of pattern $n_w(\cdot)$ of the sequence $s(t)$ is defined by:

$$n_w(s) = N^{-1} \sum_k u_s(k) I(k) \quad (7)$$

where $I(k)$ is a N units long sequence of ± 1 's.

The randomness measure $r_w(\cdot)$ is:

$$r_w(s) = 1 - n_w(s) = N^{-1} \sum_k u_s(k) J(k) \quad (8)$$

This definition is the same as given in [8], except that the measure has been defined using the discrete transform, whereas it had been done there through the analog Walsh transform of a stepped waveform corresponding to the binary sequence.

If $s(t)$ is a $(1,0)$ sequence, the number of non-zero $u_s(k)$ would for certain sequences be larger by one compared to when $s(t)$ is a $(1,-1)$ sequence. This is because transformation through $u_s(t)$ gives the average value of the sequence. Further the number of independent values of the transformed sequences would be less if $s(t)$ is considered a $(1,-1)$ series.

For example the Walsh transform of 1 1 1 0 0 1 1 is 6 0 -2 0 2 0 2 0 while that of 1 1 1 1 -1 -1 1 1 is 4 0 -4 0 4 0 4 0

We now discuss the result, given as Theorem 3, in [8]. It was claimed that for sequences of length 2^k , the ones with $r(s) < 2^{-kv}$ are less than 2^v in number. Here v can also represent the number of non-zero Walsh components of the sequence. This, as was pointed out, also means that most sequences of length 2^k have the randomness measure close to unity. It will be shown that Theorem 3 of [8] is valid only in a rough way. This is because Lemma of [8] is strictly true for a certain class of sequences only, and because for a finite population only a finite number of values of $r(s)$ will be attained.

The total number of sequences of length 2^k is 2^{2^k} . Let the number of $(1,0)$ sequences with i zeros be n_i . Then

$$n_1 = \binom{2^k}{1}$$

$$\text{Also } 2^{2^k} = \sum_{i=0}^{2^k} n_i \quad (9)$$

It can also be shown easily that for i odd, the randomness measure is always unity. (The number of sequences with $r(s) = 1$ is

$$\sum_{i \text{ odd}} n_i \quad (10)$$

$$\text{But } \sum_{i \text{ odd}} n_i = \sum_{i=1,3,5,\dots} \binom{2^k}{i} = 2^{2^k-1} \quad (11)$$

Therefore exactly one half of the population of sequences of a certain length is completely random. Now $v = 2^k - 1$ gives us the number of sequences that are not completely random and this equals 2^{2^k-1} . Theorem 3 of [8], which agrees with our result. Nevertheless for other values of v there might be some variation from the bound 2^v . Thus for $v = 2$, the theorem gives the number of such sequences as 4. However, such sequences would be at least 2^k in number, being the Walsh functions of that length. To sum up exactly half the sequences are completely random, and more sequences than given by Theorem 3 of [8] have $r(s)$ close to zero. The conclusion that most of the sequences have their randomness measure close to unity, however, holds.

The reason for the efficiency of the Walsh transform in the characterization of finite binary sequences stems from the following. The set of all binary sequences of length 2^k forms an Abelian group with addition modulo 2 as the group operation. The natural basis for a 2^k -dimensional vector space corresponding to this group is the set of Walsh functions. Walsh functions may be derived from the character group of the dyadic group. Therefore preference for Walsh functions in defining a randomness measure seems justified. The number of non-zero Walsh components gives the number of directions in this coordinate system in which the sequence has components. From the topology of a 2^k -dimensional space it can be deduced that most points on a 2^k -dimensional cube will not lie at the ends of the basis vectors.

Legendre Transform Characterization

The discrete Legendre transform [12] is one of the many discrete orthogonal sequences sets. It has been singled out for discussion since, unlike some other orthogonal sets, it has not been often used for the classification of sequences. It also has the advantage of orthogonality over any number of points.

A discrete Legendre polynomial set $\{d_j(t, I)\}$ of length I and degree 1 with t as the parameter ($t = 0, 1, \dots, I-1$) is defined by:

$$d_j(t, I) = \sum_{k=0}^j (-1)^k \binom{j}{k} \binom{t+k}{k} \frac{t^{(k)}}{(I-1)^k} \quad (12)$$

where $t^{(k)}$ is the so-called backward factorial function of order k :

$$t^{(k)} \triangleq t(t-1)\dots(t-k+1) \quad (13)$$

and $\binom{x}{j}$ is the binomial coefficient defined by:

$$\binom{x}{j} = \frac{x!}{j! (x-j)!} \quad (14)$$

The following properties can now be easily verified:

$$(i) \sum_{t=0}^{I-1} d_j(t, I) d_k(t, I) = 0, \quad j \neq k$$

$$(ii) \sum_{t=0}^{I-1} [d_j(t, I)]^2 = \frac{(I+1)(j+1)}{(2^{j+1})(j-1)} = M_j$$

and

$$(iii) \sum_{t=0}^{L-1} f(t;k) d_i(t,L) = 0, \\ 0 \leq k < 1$$

where $f(t;k)$ is any polynomial of degree k .

Any sequence $s(t)$, of length L , can be written in terms of the discrete Legendre polynomials:

$$s(t) = \sum_{i=0}^{L-1} y(i) d_i(t,L) \quad (15)$$

Multiplying both sides by $d_j(t,L)$ and summing over t after changing the order of summation, we obtain:

$$y(j) = \sum_{t=0}^{L-1} s(t) d_j(t,L) \quad (16)$$

The sequence $y(j)$ is called the discrete Legendre transform of the sequence $s(t)$.

The transformation may be done directly, or otherwise a Legendre matrix may be constructed first, which operating on the given sequence, gives the Legendre transformed sequence. The Legendre matrix may be written down using the definition (12). For $L = 8$, we thus obtain an 8th order matrix, which is:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 5/7 & 3/7 & 1/7 & -1/7 & -3/7 & -5/7 & -1 \\ 1 & 1/7 & -3/7 & -5/7 & -5/7 & -3/7 & 1/7 & 1 \\ 1 & -5/7 & -1 & -3/7 & 3/7 & 1 & 5/7 & -1 \\ 1 & -13/7 & -3/7 & 9/7 & 9/7 & -3/7 & -13/7 & 1 \\ 1 & -23/7 & 17/7 & 15/7 & -15/7 & -17/7 & 23/7 & -1 \\ 1 & -5 & 9 & -5 & -5 & 9 & -5 & 1 \\ 1 & -7 & 21 & -35 & 35 & -21 & 7 & -1 \end{bmatrix}$$

with M_i as: 8, 24/7, 24/7, 264/49, 88/7, 312/7, 1264, 3432.

As an example the Legendre transformed sequence of

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 4 & 8/7 & 12/7 & -12/7 & -2/7 & -6/7 & 6 & 14 \end{bmatrix}$$

Now pattern classification using

discrete Legendre polynomials consists of the following: Obtain the Legendre transformed sequence $y(i)$. Compute $v_s(i)$ as given below:

$$v_s(j) = \begin{cases} 1 & \text{if } \frac{y(j)}{(j!)^k} \geq 1/k \\ 0 & \text{if } \frac{y(j)}{(j!)^k} < 1/k \end{cases} \quad (17)$$

where k is an appropriately chosen constant. Thus k could be taken as equal to 1. Then the randomness measure via the Legendre transform, $r_1(L)$ is

$$r_1(s) = L^{-1} \sum_j v_s(j) I(j) \quad (18)$$

$I(j)$ is a L unit long sequence of +1's.

For 1 1 1 0 0 0 0 1, for instance, $r(s)$ by this reckoning (for $k=L$) turns out to be 0.875.

The procedure using Legendre transform is quite cumbersome compared to the one using Walsh transform, since we must each time compute the Legendre matrix afresh, which additionally is only orthogonal and not orthonormal. However its advantage lies in its applicability without constraint on the sequence length, whereas for the procedure using Walsh transforms the length L must be of the form 2^k .

In (17) $v_s(i)$ has been considered zero for normalized $v(j) < 1/k$ (k suitably large) since for such values there hopefully will be no problem in fixing the original sequence correctly on inverse Legendre transformation.

The Legendre transform characterizations of binary sequences gives most of them as being completely random. However, except for the sequence of +1's, no other binary sequence would correspond to any of the basis vectors of the space spanned by the Legendre functions. Also the randomness measure is a function of the constant k , whose choice to a great extent is arbitrary. Therefore interpretation of measure obtained through this transform is more difficult than through Walsh transform.

Conclusions

This paper is an extension of the work reported earlier by the author [8] on classification of binary sequences. An attempt has been made to show the superiority of the method employing Walsh transforms to other techniques of classification. To a degree the measure using Walsh transforms appears to give results similar to the computation -

complexity approach. Since the computation-complexity approach, further, is based on the information content of a sequence, our measure also lends itself to a similar interpretation.

References

- 1 R.J. Solomonoff, "A formal theory of inductive inference", Information and Control, vol.7, pp.1-22 and 224-254, March and June 1964.
- 2 A. Kolmogorov, "On tables of random numbers", Sankhya, ser.1, pp.369-376, December 1963.
- 3 A. Kolmogorov, "Three approaches to the definition of the concept 'quantity of information'", Prob. Peredachi Inform., vol.1, pp.7-11, 1965.
- 4 G.J. Chaitin, "On the length of programs for computing finite binary sequences", J. ACM, vol.13, 547-569, October 1966.
- G.J. Chaitin, "On the difficulty of of computations", IEEE Trans. Information Theory, vol.17-16, pp.1-5, January 1970
- 6 P. Martin-Löf, "The definition of random sequences", Information and Control, vol.9, pp.602-619, 1966.
- 7 T.I. Fine, "On the apparent convergence of relative frequency and its applications", IEEE Trans. Information Theory, vol.17-16, pp.251-257, May 1970.
- 8 S.C. Kak, "Classification of random binary sequences using Walsh-Fourier analysis", IEEE Trans. Electromagnetic Compatibility, vol.EMC-12, pp.74-77, August 1971.
- 9 S.C. Kak, "Binary sequences and redundancy", to be published.
- 10 A. Church, "On the concept of a random sequence", Bull. Am. Math. Soc., vol.46, pp.120-125, Feb. 1940.
- 11 R. von Mises, Mathematical Theory of Probability and Statistics. New York: Academic, 1964, pp.9-12.
- 12 K. Popper, The Logic of Scientific Discovery. London: Hutchinson, 1959.
- 13 F. Morrison, Introduction to Sequential Smoothing and Prediction. New York: McGraw-Hill, 1969.

ON MATRICES WITH WALSH FUNCTIONS AS THE EIGENVECTORS

Subhash C. Kak
Department of Electrical Engineering
Indian Institute of Technology
New Delhi-29, India

Abstract

The properties of Hadamard and related matrices like circulant and back-circulant matrices have been studied extensively. This paper presents a new relative of the back-circulant matrix, called the bicirculant matrix, and discusses its properties. The eigenvectors of the bicirculant matrix have been shown to be the discrete Walsh functions. The eigenvalues are obtained by finding the scalar product of any row of the bicirculant matrix with the various Walsh functions. Some particular bicirculant matrices have also been analyzed. Among these is a stochastic matrix which could have applications to statistics. The importance of bicirculant matrices to the regular representations of abstract dyadic groups has also been discussed.

Introduction

Hadamard, skew-Hadamard, circulant and back-circulant matrices have been discussed by various authors [1] - [9]. An Hadamard matrix $H = (h_{ij})$ is a

matrix of order n , all of whose elements are $+1$ and -1 and which satisfies

$$HH^T = nI, \quad I \text{ is the identity matrix.}$$

This means for an Hadamard matrix the row and the column vectors are orthogonal. An Hadamard matrix $H = [I]$ is called a skew-Hadamard if $[I]^T = -I$.

A set of elements $D = \{x_1, x_2, \dots, x_k\}$

will be said to generate a circulant $(1, -1)$ matrix $A = (a_{ij})$ if $a_{ij} =$

$a_{1, j-i+1} = 1$ when $j-i+1 \in D$ (all numbers modulo k) and -1 otherwise. A back-circulant matrix $A = (a_{ij})$ of order k has $a_{ij} = a_{1, j, i-j}$ where $1+j$ and $i-j$ are reduced modulo k .

In this paper we define a relative of the back-circulant matrix which we christen the bicirculant matrix. The eigenvectors of this matrix turn out to be the discrete Walsh functions. Various properties and some possible applications are also discussed.

Bicirculant Matrices

Definition: A bicirculant matrix $B = (b_{ij})$ of order $n = 2^k$ is defined by:

$$(i) \quad b_{1,j} = b_{j,1} = b_{n2^{-m}-1+1, n2^{-m}-j+1}; \quad 1, j \leq n2^{-m}$$

$$(ii) \quad b_{1, 2^{l-1}+j} = b_{b_{j, 2^{l-1}+1}}; \quad 1, j \leq 2^l$$

Example: A fourth-order bicirculant matrix B_4 is given below:

$$B_4 = \begin{bmatrix} b_1 & b_2 & b_3 & b_4 \\ b_2 & b_1 & b_4 & b_3 \\ b_3 & b_4 & b_1 & b_2 \\ b_4 & b_3 & b_2 & b_1 \end{bmatrix}$$

An eighth-order bicirculant matrix B_8 will similarly be:

$$B_8 = \begin{bmatrix} B_4 & B'_4 \\ B'_4 & B_4 \end{bmatrix}$$

where B_4 and B'_4 are two distinct fourth order bicirculant matrices.

We notice that a bicirculant matrix is symmetrical about both of its diagonals. Further, such a matrix can be shown to be constructed from second-order sub-matrices which are all symmetrical about both diagonals. This gives us a simple procedure to arrange the elements of such a matrix. In other words:

Lemma 1

Let B_{ij} , $(i, j = 1, \dots, N)$ satisfy the conditions on b_{ij} in the above definition. Further, let B_{ij} themselves be bicirculant matrices of order M , then $B = (B_{ij})$ is also a bicirculant matrix of order NM .

Lemma 2

If B and C are bicirculant matrices of the same orders, so is their product BC. Generalizing any $B^m C^m$ would also be a bicirculant matrix.

Proof: Since the elements of a bicirculant matrix circulate in pairs, and are symmetric about both the main diagonals, the product matrix will retain these properties. Also it follows that

Theorem 3

Bicirculant matrices commute. Or $BC = CB$.

Lemma 4

The product matrix BH , where H is an Hadamard matrix, has orthogonal columns. On the other hand the matrix HA has orthogonal rows.

Proof: Any Hadamard matrix of order 2^n can be shown to be built up from Hadamard sub-matrices of lower orders. These sub-matrices have the same sign alternations as would be obtained in a Hadamard matrix of that order. Thus if H is a Hadamard matrix of order 32 :

$$H = \begin{bmatrix} H_1 & H_1 & H_1 & H_1 \\ H_1 & -H_1 & H_1 & -H_1 \\ H_1 & H_1 & -H_1 & -H_1 \\ H_1 & -H_1 & -H_1 & H_1 \end{bmatrix}$$

where H_1 is another Hadamard matrix of order 8. Or H is seen to be a Kronecker (or direct) product of two Hadamard matrices of orders 8 and 4. This is generally expressed as the result that the Kronecker product of two Hadamard matrices of orders M and N is another Hadamard matrix of order MN. Therefore, using Lemma 1, the product BH could be expressed as:

$$\begin{bmatrix} B_1 & B_2 & B_3 & B_4 \\ B_2 & B_1 & B_4 & B_3 \\ B_3 & B_4 & B_1 & B_2 \\ B_4 & B_3 & B_2 & B_1 \end{bmatrix} \begin{bmatrix} H_1 & H_1 & H_1 & H_1 \\ H_1 & -H_1 & H_1 & -H_1 \\ H_1 & H_1 & -H_1 & -H_1 \\ H_1 & -H_1 & -H_1 & H_1 \end{bmatrix}$$

where the bicirculant matrix B has also been expressed as being built out of submatrices B_1 's. Carrying out the above multiplication, and repeating the procedure for each element of the product BH , shows the correctness of the first part of Lemma 2. The second part is proved similarly.

Combining the results of the two

parts of Lemma 2 we straightaway make the transition to the following result:

Theorem 5

B is diagonalized with the Hadamard matrix as the modal matrix. Mathematically:

$$\frac{1}{N} (HBH) = D$$

where D is the diagonalized (spectral) matrix and N the order of H and B.

Remark: We assume that the Hadamard matrices being used above are symmetric.

Since $\frac{1}{N} (H)$ is the modal matrix, its columns would be the eigenvectors of B. However the columns of H are the discrete Walsh functions in a natural ordering. Therefore the bicirculant matrix B has the Walsh functions as its eigenvectors.

Theorem 6

The i th eigenvalue of B, λ_i , is given by:

$$\lambda_i = b_{11} h_{11}$$

Proof: The product matrix BH equals $\lambda_k h_{1k}$. On pre-multiplication with the modal matrix $\frac{1}{N} (H) = h_{1j}/N$, the spectral matrix $D = (d_{1j})$ becomes $d_{1j} = \lambda_1 I$. Or the eigenvalues of B are the

scalar products of the first row of B with various columns of H. Thus if the elements of the first row of B are b_1, b_2, b_3, b_4 , the four eigenvalues are:

$$\lambda_1 = b_1 + b_2 + b_3 + b_4$$

$$\lambda_2 = b_1 - b_2 + b_3 - b_4$$

$$\lambda_3 = b_1 + b_2 - b_3 - b_4$$

$$\lambda_4 = b_1 - b_2 - b_3 + b_4$$

This implies that the eigenvalues of B would be non-zero only if

$$b_{1j} h_{1j} \neq 0 \text{ for all } i.$$

Remark: Let the sequence b_{1j} be converted into an analog waveform in the following manner [10]. For a sequence n units long, divide the interval $[-\frac{1}{2}, \frac{1}{2})$ into n equal parts. Let the magnitude at the first part, i.e., $[-\frac{1}{2}, (2-r)/2n)$, equal b_{11} , that of the second b_{12} and so on. The resulting stepped waveform is defined as the complementary analog wave (CAW) $b_1^*(t)$.

The above result can be expressed alternatively as:

The eigenvalues of B are non-zero if the Walsh-Fourier transform of $b_1(t)$ is non-zero over the sequency spread $[-2^{n-1}, 2^{n-1}]$, where 2^n is the order of B.

Bicirculant Matrices and Regular Representation

We study some particular bicirculant matrices which have applications to the theory of dyadic groups and their representation in linear spaces.

A bicirculant matrix of order 2^n will lead to 2^n independent (1,0) matrices if $b_1 = \delta_{1j}$ with $j = 1, 2, 3, \dots, 2^n$ in turn. Thus for a fourth order matrix we get the four independent matrices as:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Further since all these matrices satisfy the conditions of Theorem 6, their eigenvalues are non-zero, and they have Walsh functions as their eigenvectors. It will now be shown that these matrices are extremely important in the representation of dyadic groups.

An Abelian (or commutative) group can also be represented as a matrix set in a linear space. Let G be an Abelian group of n elements

$$a_0, a_1, \dots, a_2, \dots, a_{n-1}$$

with group operation \square . Then

$$\begin{aligned} a_1 \square a_j &= a_k \\ &= a_j \square a_1 \end{aligned}$$

One can now construct an n-dimensional vector space L with a_1 as the basis vectors. A representation of the group G in the linear vector space L is a correspondence between the group

elements of G and certain matrices of L such that the product of the matrices preserves the group operation. Thus if:

$$a_1 \square a_k = a_m \text{ and } a_j \mapsto M(a_j)$$

where the element a_j maps into the matrix $M(a_j)$ in L, then

$$M(a_1) M(a_k) = M(a_m)$$

One important representation of an Abelian group is the regular representation. This is defined as the set of matrices of real numbers which correspond to the elements of G such that the product of two such matrices corresponds to the element of G given by the group multiplication. In short, it represents a homomorphism of an abstract group into a group of real matrices.

Finite dyadic groups have 2^k elements. A simple abstract dyadic group consists of two elements 1 and d where $d^2 = 1$. A realization of primary importance for such groups is the binary n-tuplets taking on the values 0 and 1 at each position of n-tuplets with the group operation being addition modulo 2. In Rosenbloom [1] it has been shown that the regular representation for a binary n-tuplet realization of abstract dyadic groups are the matrices listed in the beginning of this section, i.e., matrices obtained by putting $b_1 = \delta_{1j}$ in turn for $j = 0, 1, \dots, n-1$ in B. Now since the Walsh functions are a natural bases for a 2^n -dimensional vector space corresponding to a 2^n element Abelian group with addition modulo 2 as the group operation, they turn out to be the eigenvectors of the above-mentioned regular representation. In this paper, using the property of bicirculant matrices, this result has been obtained otherwise.

The group property of the regular representation will be explicitly stated. Let R_1^i , $i = 0, 1, \dots, n-1$ be the regular representation in a n-dimensional vector space. The matrix R_1^i is obtained by putting $b_j = \delta_{1j}$ in the general bicirculant matrix B. Thus

$$R_2^4 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

We can now easily write the following result.

Theorem 7

For R_1^n defined as above

$$R_1^n R_j^n = R_1^n \oplus_j$$

where $1 \oplus j$ represents addition modulo 2 after 1 and j have been put in a binary representation.

Examples.

$$i) R_0^4 R_3^4 = R_3^4$$

since $0 \ 0 \oplus 1 \ 1 = 1 \ 1$

$$ii) R_1^4 R_1^4 = R_0^4$$

since for any 1 addition modulo 2 yields a sequence of zeros. Thus

$$\begin{array}{ccccccc}
 1 & 0 & 0 & 1 & 1 & 1 & 0 \\
 \oplus & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\
 \hline
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
 \end{array}$$

$$iii) R_1^4 R_3^4 = R_2^4$$

Applications to Stochastic Matrices

In case $\sum_j b_{1j} = 1$, the bicircular matrix B can be used as a stochastic matrix. Further if all $b_{1j} \neq 0$, B would represent the transition matrix of a regular finite Markov chain [12]. Because of its structure it would represent a perfectly symmetrical finite state scheme. Therefore the limiting distribution would contain identical entries. If B is the transition matrix corresponding to a n -state chain then each entry in the limiting distribution would be $1/n$.

Theorem 8

If $b_{1j} \neq 0$ and $\sum_j b_{1j} = 1$, then

$$\lim_{r \rightarrow \infty} B^r = \frac{1}{n} U$$

where U is a $n \times n$ matrix with all $b_{1j} = 1$.

Discussion and Conclusion

In this paper we define and study bicirculant matrices. Their intimate relationship with the regular representation of abstract dyadic groups is discussed in addition to possible applications as stochastic matrices.

Bicirculant matrices can also be

used to analyze data. However any such analysis would be equivalent to expanding the data in a weighted series of Walsh functions.

References

- 1 Marshall Hall Jr., Combinatorial Theory, Waltham, Mass.: Blaisdell, 1967
- 2 J. Williamson, "Hadamard's determinant theorem and the sum of four squares", Duke Math. J., vol. 11, pp. 65-81, 1944
- 3 R.E.A.C. Paley, "On orthogonal matrices", J. Math. Phys., vol. 12, pp. 311-320, 1933
- 4 W.K. Pratt, J. Yane and H.C. Andrews, "Hadamard transform image coding", Proceedings IEEE, vol. 57, pp. 58-68, January 1969
- 5 Jennifer Wallis, "(v, k, λ) configurations and Hadamard matrices", J. Australian Math. Soc., vol. 11, pp. 297-309, 1970
- 6 Jennifer Wallis, "Some (1, -1) matrices", J. Combinatorial Th., vol. 10, pp. 1-11, February 1971
- 7 J. Hadamard, "Resolution d'une question relative aux determinants", Bull. Sci. Math., ser. 2, vol. 17, pt. 1, pp. 240-246, 1893
- 8 F.J. Ryser, Combinatorial Mathematics, New York: Wiley, 1963
- 9 L.R. Welch, "Walsh functions and Hadamard matrices", Proceedings Symposium on Applications of Walsh Functions, pp. 162-165, April 1970
- 10 S.C. Kak, "Classification of random binary sequences using Walsh-Fourier analysis", Proceedings Symposium on Applications of Walsh Functions, pp. 74-77, April 1971
- 11 J.W. Rosenbloom, "Physical interpretation of the dyadic group", Proceedings Symposium on Applications of Walsh Functions, pp. 158-165, April 1971
- 12 E.B. Dynkin, Markov Processes, Vols. I and II, New York: Academic, 1965

APPROXIMATION BY WALSH POLYNOMIALS AND THE CONCEPT OF A DERIVATIVE

P.L. Butzer and H.J. Wagner
Technological University of Aachen
Aachen, Germany

1. Introduction

In a recent paper the authors [3] defined a derivative for functions f given on the dyadic group G . This derivative turned out to be a linear, closed operator, its inverse operator or integral was introduced, and the fundamental theorem of the calculus was found to hold for these two concepts. This enabled one to obtain first results on the approximation of f by the partial sums of the Walsh-Fourier series of f . The purpose of this paper is to present the definition of the derivative, this time in its setting on the unit interval $[0,1]$, as well as to give a new application, namely to establish the fundamental theorem on best approximation (in the version of [1,2] for functions f defined on $[0,1]$ by Walsh polynomials.

2. Preliminary results

Defining the Rademacher functions by

$$\phi_0(x) = \begin{cases} 1, & 0 \leq x < 1/2 \\ -1, & 1/2 \leq x < 1 \end{cases}, \quad \phi_0(x+1) = \phi_0(x),$$

$$\phi_n(x) = \phi_0(2^n x) \quad (n \in \mathbb{N} = \{1, 2, \dots\}),$$

then the Walsh functions are given by

$$(2.1) \quad \Psi_0(x) = 1, \Psi_n(x) = \phi_{n_1}(x) \dots \phi_{n_i}(x)$$

$n = 2^{n_1} + \dots + 2^{n_i}, n_1 > \dots > n_i > 0, n_j$ being integers. Let G be the dyadic group consisting of all sequences $\bar{x} = \{x_n\}_{n=1}^\infty$ such that $x_n = 0$ or 1 , the operation of G being addition modulo 2 (notation \oplus). G is related to the unit interval by the mapping $\lambda: G \rightarrow [0,1]$

$$\lambda(\bar{x}) = \sum_{n=1}^\infty 2^{-n} x_n.$$

λ does not have a single-valued inverse since the dyadic rationals (=D.R.) have two representations in G . We shall agree to take the finite expansion in that case. If μ is the inverse of λ , then, according to N.J. Fine [4], one has for all real x

$$\lambda(\mu(x)) = x - [x],$$

$[x]$ denoting the greatest integer $\leq x$. Moreover, $\mu(\lambda(\bar{x})) = \bar{x}$ for all $\bar{x} \in G$ provided $\lambda(\bar{x}) \notin$ D.R. Denoting $\lambda(\mu(x) \oplus \mu(y))$ in short by $x \oplus y$, then Fine [4] showed that

$$\int_0^1 f(x \oplus y) dx = \int_0^1 f(x) dx$$

for every fixed y provided that f is Lebesgue integrable on $[0,1]$. Furthermore, for each fixed y and for all x outside a certain denumerable set (depending on y), one has

$$(2.2) \quad \Psi_n(x \oplus y) = \Psi_n(x) \Psi_n(y).$$

More generally, (2.2) is also valid for all $(x, y) \in [0,1] \times [0,1]$ outside a denumerable set.

A finite linear combination $\sum_{k=0}^{n-1} c_k \Psi_k(x)$ (c_k being complex numbers) is called a Walsh polynomial of degree n , the set of such polynomials of degree $\leq n$ being denoted by P_n . If $p_m \in P_{2^n}$, $h \in [0, 2^{-n})$, then, outside the above denumerable set,

$$(2.3) \quad p_m(x \oplus h) = p_m(x).$$

Denoting by $L^p(0,1)$, $1 \leq p < \infty$, the set of all functions f of period 1 which are p th power integrable with norm $\|f\|_p = \left(\int_0^1 |f(x)|^p dx \right)^{1/p}$, it is known

that the Walsh-system $\{\psi_k(x)\}_{k=1}^{\infty}$ is closed with respect to the space $L^p(0,1)$, i.e.,

$$(2.4) \quad f^{\wedge}(k)=0, \quad k \in P, \Rightarrow f(x)=0 \quad \text{a.e.},$$

the Walsh-Fourier coefficients of f being

$$f^{\wedge}(k) = \int_0^1 f(u) \psi_k(u) du \quad (k \in P = \{0, 1, 2, \dots\}).$$

By (2.2) for each real y and $k \in P$

$$(2.5) \quad \int_0^1 f(y \oplus u) \psi_k(u) du = \psi_k(y) f^{\wedge}(k).$$

Defining the convolution of $f \in L^p(0,1)$ and $g \in L^1(0,1)$ by

$$\begin{aligned} (f * g)(x) &= \int_0^1 f(x \oplus u) g(u) du \\ &= \int_0^1 f(u) g(x \oplus u) du, \end{aligned}$$

it is also known that $f * g$ exists for almost all real x , $f * g \in L^p(0,1)$ and

$$(2.6) \quad \|f * g\|_p \leq \|f\|_p \|g\|_1.$$

Moreover, it is obvious that for $f \in L^p(0,1)$

$$|f^{\wedge}(k)| \leq \|f\| \quad (k \in P),$$

and the convolution theorem states that for $f, g \in L^1(0,1)$

$$(2.7) \quad (f * g)^{\wedge}(k) = f^{\wedge}(k) \cdot g^{\wedge}(k) \quad (k \in P).$$

Finally, if $f_n, f \in L^p(0,1)$, $n \in \mathbb{N}$, then for all $k \in P$

$$(2.8) \quad \lim_{n \rightarrow \infty} \|f_n - f\| = 0 \Rightarrow \lim_{n \rightarrow \infty} f_n^{\wedge}(k) = f^{\wedge}(k)$$

3. The Derivative and its properties

Definition

If for $f \in L^p(0,1)$ there exists $g \in L^p(0,1)$ such that

$$(3.1) \quad \lim_{m \rightarrow \infty} \frac{1}{2^m} \sum_{j=0}^m 2^j [f(\cdot) \psi_{2^j-1}(\cdot) - g(\cdot)] = 0,$$

then g is called the strong derivative of f , denoted by $D^{[1]}f$. For $r=2,3,\dots$ the r th strong derivative of $f \in L^p(0,1)$ is defined successively by

$$D^{[r]}f = D^{[1]} D^{[r-1]}f.$$

Note that this definition differs from that given by J.E. Gibbs [5] essentially in the sense that the factor 2^j is replaced by 2^{-j} ; however, Gibbs' definition is taken in the pointwise sense.

$D^{[r]}$ is a linear operator with the property

Proposition 3.1

$$(3.2) \quad D^{[r]} \psi_n = n^r \psi_n \quad (n \in P, r \in \mathbb{N}).$$

Using (2.2), the proof for $r=1$ follows readily from the identity ($n \in P$)

$$(3.3) \quad \sum_{j=0}^{\infty} 2^j [1 - \psi_n(2^{-j-1})] = 2n.$$

To prove (3.3), note that for $m, j \in P$ $\psi_m(2^{-j-1}) = \psi_m(2^m 2^{-j-1}) = -1$ for $m=j$, $=1$ for $m \neq j$. Then apply (2.1).

Hence Walsh polynomials are arbitrarily often differentiable. Furthermore, (3.3) together with (2.8) implies that

$$[D^{[r]}f]^{\wedge}(k) = k^r f^{\wedge}(k) \quad (k \in P)$$

under the existence of $D^{[r]}f \in L^p(0,1)$. It is obvious that the strong derivative of a constant vanishes and, conversely, $D^{[1]}f = 0$ implies $f = \text{const.}$ in view of (2.5), (3.3), (2.8) and (2.4).

Of importance is the function $W_r(x)$ defined by (see Watari [7])

$$W_r^{\wedge}(k) = \begin{cases} 1, & k = 0 \\ k^{-r}, & k \in \mathbb{N} \end{cases} \quad (r \in \mathbb{N}).$$

Having the theory developed in Butzer-Wagner [3] at our disposal, it can be shown by parallel methods that $W_r(x) \in L^1(0,1)$. Moreover, if $f, g \in L^p(0,1)$ such that $k^r f^{\wedge}(k) = g^{\wedge}(k)$, $k \in \mathbb{P}$, $r \in \mathbb{N}$, then

$$f = W_r * g + f^{\wedge}(0).$$

Theorem 3.1

Let $f \in L^p(0,1)$ such that $f^{\wedge}(0) = \int_0^1 f(x) dy = 0$.

a) If there exists $D^{[r]} f \in L^p(0,1)$ for some $r \in \mathbb{N}$, then

$$(W_r * D^{[r]} f) = f$$

b) $D^{[r]} (W_r * f) = f$.

This theorem may be regarded as the fundamental theorem of the calculus for our concept of a derivative. In this sense, our operator of integration I is given by

$$If = (W_1 * f) = \int_0^1 f(x \oplus u) W_1(u) du.$$

It is obvious that I is linear and continuous. Or the other hand, it can be shown that the operator $D^{[r]}$ of differentiation is closed on $LP(0,1)$.

All of the results established in Butzer-Wagner [3], including the applications, carry over from the dyadic group G to the interval $[0,1]$. The following new applications are proved in detail.

4. Applications to Approximation by Walsh Polynomials

In the notation of G.W. Morgenthaler [6] define for $f \in L^p(0,1)$, $\alpha > 0$

$$\text{Lip } \alpha(W) = \{f; \|f(\cdot) - f(\cdot \oplus h)\| = O(h^\alpha), h \rightarrow 0\}$$

$$\omega_W(f; \delta) = \sup_{0 < h < \delta} \|f(\cdot) - f(\cdot \oplus h)\|,$$

$$E_n(f) = \inf_{p_n \in P_n} \|f - p_n\|,$$

the latter quantity being called best approximation of f by Walsh polynomials of degree n in $L^p(0,1)$ -space. It is known that there exists a Walsh polynomial p_n^* ($= p_n^*(f)$) of best approximation to $f \in L^p(0,1)$ for which $E_n(f) = \|f - p_n^*\|$.

According to Morgenthaler and Watari [8] it is known that for $\delta \rightarrow 0$

$$(4.1) f \in \text{Lip } \alpha(W) \Leftrightarrow \omega_W(f; \delta) = O(\delta^\alpha),$$

$$(4.2) E_{2^n}(f) \leq \omega_W(f; \frac{1}{2^n}) \leq 2 E_{2^n}(f),$$

respectively. Analogous to [3] we have

Proposition 4.1

If $D^{[r]} f$ exists and belongs to $L^p(0,1)$, then

$$\omega_W(f; \frac{1}{2^n}) = O(\frac{1}{2^{nr}} \omega_W(D^{[r]} f; \frac{1}{2^n})).$$

Of fundamental importance are the Bernstein and Jackson-type inequalities given by

Proposition 4.2

a) For $p_n \in P_n$ one has

$$\|D^{[r]} f - p_n\| \leq A n^r \|p_n\| \quad (n \in \mathbb{P}, r \in \mathbb{N}).$$

b) If $D^{[r]} f$ exists and belongs to $L^p(0,1)$, then

$$E_n(f) \leq B n^{-r} \|D^{[r]} f\|.$$

Here A and B are constants independent of n , p_n , f .

Proof

If n has the representation (2.1), then $p_n \in P_{n_1+1}$. By (2.3) one has

$$\begin{aligned} & \frac{1}{n} \|D^{[1]} p_n\| \\ &= \frac{1}{n} \left\| \frac{1}{2} \sum_{j=0}^{\infty} 2^j [p_n(\cdot) - p_n(\cdot \oplus 2^{-j-1})] \right\| \\ &< \frac{1}{n} \sum_{j=0}^{n_1} 2^j \|p_n\| \leq \|p_n\| \sum_{j=0}^{n_1} 2^{j-n_1} < 2 \|p_n\|. \end{aligned}$$

The proof of a) now follows by induction.

Under the hypotheses of b) the assertion of b) follows immediately by (4.2) and Prop. 4.1.

The following, which is the fundamental theorem on best approximation by Walsh polynomials, shows the usefulness of our derivative concept.

Theorem 4.1

If $f \in L^p(0,1)$, the following statements are equivalent:

- (i) $D^{[r]} f \in \text{Lip } \alpha(W) \quad (\alpha > 0),$
- (ii) $\omega_W(D^{[r]} f; \frac{1}{n}) = O(n^{-\alpha}),$
- (iii) $E_n(f) = O(n^{-\alpha-r}),$
- (iv) $D^{[v]} f \in L^p(0,1), \quad 0 < v < r,$
 $\|D^{[v]} f - D^{[v]} p_n^*\| = O(n^{+v-\alpha-r}),$
- (v) $\|D^{[1]} p_n^*\| = O(n^{1-\alpha-r}) \quad (0 < \alpha + r < 1).$

Proof

The equivalence (i) \Leftrightarrow (ii) follows by Morgenthaler [6] (who considered case $r = 0$; see (4.1)). The implication (ii) \Rightarrow (iii) follows by (4.2) and Prop. 4.1.

To prove (iii) \Rightarrow (iv) first note that (iii) implies $\|U_k^*\| = O(2^{-k(\alpha+r)})$,

$$\text{where } U_k^* = \begin{cases} p_{2^k}^* - p_{2^{k-1}}^*, & k=2,3,\dots, \\ p_1^*, & k=1. \end{cases}$$

Therefore by Prop. 4.2 a) for $k \in \mathbb{N}$

$$(4.3) \quad \|D^{[v]} U_k^*\| = O(2^{k(v-\alpha-r)}).$$

Now the sum $\sum_{k=1}^m U_k^*$ converges in $L^p(0,1)$ -norm to f . Since for $0 < v < r$

$$\left\| \sum_{k=1}^m D^{[v]} U_k^* \right\| < C \sum_{k=1}^m 2^{k(v-\alpha-r)} < \infty,$$

there exist functions $g_v \in L^p(0,1)$ with

$$\lim_{m \rightarrow \infty} \left\| \sum_{k=1}^m D^{[v]} U_k^* - g_v \right\| = 0.$$

Since the operators $D^{[v]}$ are closed, $D^{[v]} f = g_v$ for $0 < v < r$. By (4.3) we have

$$\begin{aligned} \|D^{[v]}(f - p_{2^m}^*)\| &= \|D^{[v]} f - \sum_{k=1}^m D^{[v]} U_k^*\| \\ &< \left\| \sum_{k=m+1}^{\infty} D^{[v]} U_k^* \right\| = O(2^{m(v-\alpha-r)}). \end{aligned}$$

Now for $2^m \leq n < 2^{m+1}$ one has by (4.3) $\|D^{[v]}(p_n^* - p_{2^m}^*)\| = O(n^{v-\alpha-r})$. Combining the two inequalities yields (iv).

(iv) \Rightarrow (ii) by (4.2). It remains to show that (iii) \Leftrightarrow (v). This follows along standard lines, compare Butzer-Scherer [1, p.118 f, or 2].

In contrast to the situation for classical Fourier series, the assertion (ii) need only be formulated for the modulus of continuity of the first difference.

The results of this paper may also be established for the space $L^\infty(0,1)$, taking into account slight modifications.

The contribution of the second-named author was supported by the Minister für Forschung des Landes Nordrhein-Westfalen - Landesamt für Forschung.

References

- [1] P.L. Butzer - K. Scherer: Über die Fundamentalsätze der klassischen Approximationstheorie in abstrakten Räumen, in: "Abstract Spaces and Approximation" (Proceedings of the Oberwolfach Conference 1968, P.L. Butzer and B. Sz. Nagy, Eds.), ISNM, Vol. 10. (Basel 1969), 113 - 125.
- [2] P.L. Butzer - K. Scherer: Jackson and Bernstein-type inequalities for families of commutative operators in Banach spaces, J. Approximation Theory 5 (1972) (in print).
- [3] P.L. Butzer - H.J. Wagner: Walsh-Fourier series and the concept of a derivative, Applicable Analysis (in print).
- [4] N.J. Fine: On the Walsh functions, Trans. Amer. Math. Soc., 65 (1949), 372 - 414.
- [5] J.E. Gibbs - B. Ireland: Some generalizations of the logical derivative, NPL : DES Report No. 8, 1971.
- [6] G.W. Morgenthaler: On Walsh-Fourier series, Trans. Amer. Math. Soc., 84 (1957), 472 - 507.
- [7] C. Watari: Multipliers for Walsh-Fourier series, Tôhoku Math. J. (2) 16 (1964), 239 - 251.
- [8] C. Watari: Approximation of functions by a Walsh-Fourier series, Proceedings of the Symposium and Workshop on Applications of Walsh Functions, Naval Research Laboratory, Washington D. C., 1970, 166 - 169.

A NEW EYE ON WALSH FUNCTIONS

by

Claude CARDOT
Senior Engineer

CIT-ALCATEL - Marcoussis(91)-FRANCE

Abstract

We introduce two new algebraic representations for Walsh functions. The sign of a trigonometrical product allows to compute Fourier spectrum and to classify Walsh functions into diaphonic equivalence classes. A polynomial representation gives the key to the convolution products of any two Walsh functions. A theoretical consequence is the fact that 5^k distinct convolution products only exist in a class containing 2^k Walsh functions. Practical consequences are formulas for the influence of limited bandwidth, additive noise and synchronization error in the operation of a Walsh carrier multiplex.

Notations

The definition interval for Walsh functions will be : $(-\pi, \pi)$.

x will be the independent variable.
Compared to usual notations, we shall have $x = 2\pi\theta$.

$W(N, x)$ (N integer) will be the value of the Walsh function of rank N if the Walsh functions are sequency ordered and numbered from 0 to $2^k - 1$.

$W(N, i)$ (N, i integers) will be the constant value (± 1) of $W(N, x)$ on the interval number i of its definition period.

$i \oplus j$ will denote the direct sum of integers i, j .

$i \dot{+} j$ will denote the C-sum of two binary numbers with k digits, the result being a k -digit ternary number with the rules :

$$\begin{aligned} 1 \dot{+} 1 &= +1 \\ 0 \dot{+} 0 &= -1 \\ 1 \dot{+} 0 &= 0 \dot{+} 1 = 0 \end{aligned}$$

The word simple will keep its standard English meaning and the words : spectrum, Fourier series, their classical trigonometrical sense.

A Walsh carrier is an indefinitely repeated Walsh function.

The word diaphony is used here as an equivalent to "crosstalk".

Trigonometric representation

The 2^k first Walsh functions being sequency ordered from $N = 0$ to $N = 2^k - 1$, and their sign being normalized so that $W(+0, x) = +1$ for all N , we have :

$$W(i, x) \cdot W(j, x) = W(i \oplus j, x), \quad i, j \text{ integers. (1)}$$

The operation \oplus defines an Abelian group on the k binary digits integers.

Integers with one single 1 and $(k-1)$ zeroes constitute a simple generators set for this group. If no generator is repeated in a sum, direct sum is identical to normal addition for this set.

These integers are powers of 2. So that if we operate the dyadic decomposition of N :

$$(1) \quad N = \sum_j a_j \cdot 2^{j-1} \quad a_j = 0, 1 \text{ and } j=1, 2, \dots, k$$

we shall have :

$$W(N, x) = \prod_j W(2^{j-1}, x) \text{ the product being for all } j \text{ such that } a_j = 1.$$

Now, $W(1, x)$ is the first Rademacher function, and we have :

$$W(1, x) = W(2^0, x) = \text{sgn}(\sin x) = \text{sgn}(\cos(x - \pi/2))$$

$W(2, x)$ is the same function cyclically shifted and we have :

$$W(2^1, x) = \text{sgn}(\cos x), \text{ and :}$$

$$W(2^2, x) = \text{sgn}(\cos 2x)$$

$$\dots\dots\dots W(2^{k-1}, x) = \text{sgn}(\cos 2^{k-2}x)$$

all these functions being cyclically shifted even Rademacher functions, with orders from 1 to $k-1$.

The operation "sgn" commutes with multiplication, so that any Walsh function can be obtained as the sign of a trigonometrical product, defined as follows by the bits 1 in N :

- the first bit from right, if 1, gives a factor : $\sin x$,

- the second bit from right, if 1, gives a factor : $\cos x$,
- the third bit from right, if 1, gives a factor : $\cos 2x$
-
- the bit of rank k from right, if 1, gives a factor : $\cos 2^{k-2}x$.

With the a_i defined in (1), we have :

$$W(N, x) = \text{sgn} \cos a_1 (x - \pi/2) \prod_{j=2}^k \cos (a_j 2^{j-2} x) \quad (2)$$

This formula allows to compute the 2^k first Walsh function with $k+1$ storage registers only, on any computer with a cosine subroutine. k registers are needed to store the a_i and one more to effect the trigonometrical product.

To implement this formula, x shall be restricted to the mid-interval values, so that the sign extracting sequence cannot fail on a zero. Setting : $x = (2i-1)\pi/2^k$; $i = (1-2^{k-1}), \dots, 2^{k-1}$, the preceding formula becomes :

$$W(N, i) = \text{sgn} \cos(a_1 (2i-1)\pi/2^k - \pi/2) \cdot \prod_{j=2}^k \cos(a_j (2i-1) 2^{j-1} \pi/2^{k+1}) \quad (3)$$

Trigonometric series for a Walsh carrier

If we write a Walsh carrier under the form :

$$W(N, x) = \sum_{i=1}^{2^{k-1}} W(N, i) \cdot u_i(x) \quad (4)$$

$i = 1, 2, \dots, 2^{k-1}$ describing the right half of the period and the $u_i(x)$ being 2^{k-1} orthogonal functions obtained by infinite repetition of :

- a function equal to +1 on intervals $+i$ and $(i-1)$ of fundamental period, if N is even, and zero elsewhere,
- a function equal to +1 on interval $+i$ and to -1 on interval $(i-1)$ if N is odd, and zero elsewhere.

The trigonometric series for $u_i(x)$ is :

even N case : $u_i(x) = \sum_{n=1}^{\infty} \frac{2}{\pi n} \cdot \left[\sin \frac{\pi}{2^{k-1}} n i - \sin \frac{\pi}{2^{k-1}} n (i-1) \right]$

odd N case :

$$u_i(x) = \sum_{n=1}^{\infty} \frac{2}{\pi n} \sin nx \left[\cos \frac{\pi}{2^{k-1}} n (i-1) - \cos \frac{\pi}{2^{k-1}} n i \right]$$

and, reporting this in (4), we derive :

Even case : ($N = 2s$) :

$$W(2s, x) = \sum_{n=1}^{\infty} \frac{2}{\pi n} (\cos nx) \sum_{i=1}^{2^{k-1}} W(2s, i) \cdot \left[\sin \frac{\pi n i}{2^{k-1}} - \sin \frac{\pi n (i-1)}{2^{k-1}} \right] \quad (5)$$

Odd case : ($N = 2s - 1$) :

$$W(2s-1, x) = \sum_{n=1}^{\infty} \frac{2}{\pi n} (\sin nx) \sum_{i=1}^{2^{k-1}} W(2s-1, i) \cdot \left[\cos \frac{\pi n (i-1)}{2^{k-1}} - \cos \frac{\pi n i}{2^{k-1}} \right] \quad (6)$$

the coefficients $W(N, i)$ in (5) and (6) being computed from (3).

Spectra classification

Theorem 1. If k_0 is the rank of the first non-zero bit from right in the sequence, then Walsh carriers with this sequence shall have spectral lines on odd multiples of 2^{k_0} .

Proof : $\text{sgn} \cos z$ and $\text{sgn} \sin z$ have spectral lines on odd multiples of z .

Now, let, in binary notation :

$s = xyz10$; xyz being any binary digits ($k_0 = 2$ in this case)

$2s = xyz100$; $W(2s, x) = \text{sgn}[Y(x) \cdot \cos 2^{k_0-1} x]$ by (2)

$2s-1 = xyz011$; $W(2s-1, x) = \text{sgn}[Y(x) \cdot \sin 2^{k_0-2} x \cdot \sin 2^{k_0-1} x \cdot \sin x]$

Applying k_0 times from right to the last expression the identity : $\text{sgn}(\sin x \cdot \cos x) = \text{sgn} \sin 2x$, we see that :

$$W(2s-1, x) = \text{sgn}[Y(x) \cdot \sin 2^{k_0-1} x]$$

$Y(x)$ containing only cosines of higher frequencies, the theorem results. The cal function contains cosines of odd multiples of

k_0-1
 $2^{k_0} x$ and the sal function: sines of the same frequencies.

A result from this theorem is that if any two Walsh carriers share one Fourier component, they share them all, so that sharing one Fourier component is an equivalence relation.

The equivalence classes thus obtained can be labeled by k_0 and, for each value of k_0 , there will be two classes: one for even, and one for odd functions.

Inside each class, the functions can be sequency ordered by the bits at left of the first non-zero bit in s . Obtained rank will be named the diaphonic equivalence number, e .

It is easy to see that there are 2^k classes in the group formed by the 2^k first Walsh functions.

Table fig. 1 shows these 10 classes for the 32 first Walsh functions, with the corresponding e -ordering.

Rank N	Even functions	Sequency s	k_0	e	Odd functions	Rank N
0	cal (0, x)	0	—	—		
2	cal (1, x)	1 = 0001	1	0	sal (1, x)	1
6	cal (3, x)	3 = 0011	1	1	sal (3, x)	5
10	cal (5, x)	5 = 0101	2	2	sal (5, x)	9
14	cal (7, x)	7 = 0111	3	3	sal (7, x)	13
18	cal (9, x)	9 = 1001	4	4	sal (9, x)	17
22	cal (11, x)	11 = 1011	5	5	sal (11, x)	21
26	cal (13, x)	13 = 1101	6	6	sal (13, x)	25
30	cal (15, x)	15 = 1111	7	7	sal (15, x)	29
4	cal (2, x)	2 = 0010	2	0	sal (2, x)	3
12	cal (6, x)	6 = 0110	1	1	sal (6, x)	11
20	cal (10, x)	10 = 1010	2	2	sal (10, x)	19
28	cal (14, x)	14 = 1110	3	3	sal (14, x)	27
8	cal (4, x)	4 = 0100	3	0	sal (4, x)	7
24	cal (12, x)	12 = 1100	1	1	sal (12, x)	23
16	cal (8, x)	8 = 1000	4	0	sal (8, x)	15
		16 = 10000	5	0	sal (16, x)	31

Fig. 1 Equivalence classes for the 32 first Walsh functions. Heavy lines separate distinct classes.

Orthogonality and diaphony

Consider now the Walsh multiplex fig. 2 where $N = 2^k$ analogical samples s_0, \dots, s_{N-1} are transmitted through a line, each of them modulating a Walsh carrier $W(N, x)$ and demodulation being effected by N integrate-and-dump circuits, IT_j , fed by the corresponding Walsh carriers.

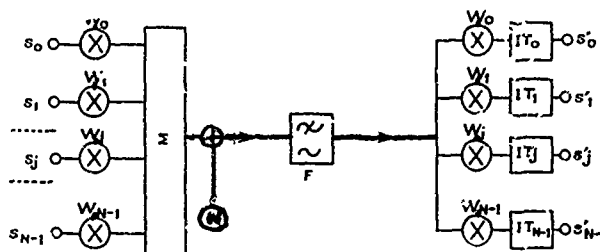


Fig. 2 Walsh multiplex

We suppose here that:

- the s_i are constant during a period of the Walsh carriers,
- there is neither synchronization error nor additive noise ($N(t) = 0$);
- the line is a rectangular low-pass filter, cutting off all spectral components with $n > n_{\max}$ and having no effect on the lower components.

The demodulated samples, s'_j , are bound to the s_i by the linear system:

$$s'_j = \sum_{i=0}^{N-1} D_{ij} \cdot s_i$$

When low-pass filtering is present, the D_{ij} terms are functions of n_{\max} , the cutoff frequency of the filter.

When there is no filtering ($n_{\max} = \infty$), the D_{ij} matrix is unitary, because of the orthonormality of the Walsh functions set.

For all practical purposes, the sign of the D_{ij} coefficients is unimportant, so that we shall neglect it in the sequel.

When a $D_{ij}(n_{\max})$ coefficient is identically zero, we say that corresponding Walsh carriers ($N = i$ and $N = j$) are absolutely orthogonal (orthogonal for any filtering).

This is the case either if i and j have not the same parity (The Fourier series are made of cosines for one of them and of sines for the other) or if the k_j values are not the same (The spectral lines are not on the same frequencies).

When two Walsh functions are not absolutely orthogonal, we have:

$$D_{ij}(r_{\max}) = \frac{1}{2} \sum_{n=1}^{n_{\max}} f_{ni} \cdot f_{nj} \quad (7)$$

for the expression of the filtering-induced diaphony between the carriers $W(i, x)$ and $W(j, x)$; as a particular case :

$$D_{ii}(r_{\max}) = \frac{1}{2} \sum_{n=1}^{n_{\max}} f_{ni}^2 \quad (8)$$

expresses the filtering-induced diaphony for the carrier $W(i, x)$, if :

$$W(i, x) = \sum_{n=1}^{n_{\max}} f_{ni} \cos nx \sin nx$$

$$W(j, x) = \sum_{n=1}^{n_{\max}} f_{nj} \cos nx \sin nx$$

Universal diaphony matrix

In preceding formulas, i and j are supposed to correspond to Walsh functions in the same equivalence class. In that case, k_0 being the index of that class, i and j can be expressed as functions of k_0 and of equivalence numbers : e, e' of $W(i, x)$, $W(j, x)$ respectively. We can denote the $D_{ij}(n_{\max})$ coefficient as :

$$D_{e, e'}^{k_0}(n_{\max}).$$

It results from Theorem 1 that classes with different values of k_0 correspond one to another by a modification of the frequency scale by the factor $2^{k_0 - k_0'}$. In particular, we have :

$$D_{e, e'}^{k_0}(n_{\max}) = D_{e, e'}^{k_0'}(n_{\max} / 2^{k_0 - k_0'})$$

That is to say, all diaphony classes are isomorphic and are described by the same matrix, with a suitable multiplication of the frequency scale. An even class being evidently isomorphic to the odd class with the same k_0 , we see that in case of rectangular filtering, all diaphonic coefficients between Walsh carriers are given by an universal matrix $D_{e, e'}(n)$ whose coefficients are functions of n , e, e' the lowpass cutoff frequency.

Moreover, in the case of another type of filter, the same general properties hold, since they rely on the spectra patterns of the Walsh functions before filtering. Of

course, the numerical values of the matrix coefficients shall no longer be given by the simple expressions (7) and (8), but by summations taking into account the phase and amplitude characteristics of the filter actually used.

Influence of Noise in a Walsh multiplex

Suppose now that, in fig 2, an additive noise source $N(t)$ is present, delivering white gaussian noise with spectral power density N from zero frequency to F .

The noise appearing in the demodulated channel s'_j will be the product of the pass-band of this channel by the noise density around zero-frequency in the demodulated output s'_j , because the integrate-and-dump circuit acts as a lowpass filter.

Demodulated noise before the integrate and dump circuit will be the power addition of all components obtained by multiplication of incident noise with each of the Fourier component of $W(j, x)$:

$$N'(t) = N(t) \times \sum_{n=1}^{n_{\max}} f_{nj} \frac{\sin 2\pi n t/T}{\cos 2\pi n t/T}$$

(T = period)

Noise density of this product at zero frequency is :

$$N'(f) = N \times \frac{1}{2} \sum_{n=1}^{n=FT} f_{nj}^2 = D_{jj}(FT) \cdot N$$

because spectral lines with $n > FT$ give transposed noise bands without component at zero frequency.

We see that noise density on the j th output is attenuated with the filtering attenuation coefficient corresponding to the upper noise frequency F , and to the j th Walsh function.

This results intuitively from the fact that the insertion of a lowpass filter with cutoff frequency F will not modify the effect of the noise.

The attenuation for the noise density being the same as the attenuation for the corresponding signal, we see that if the noise is white in the bandpass of the line, the signal to noise power ratio will be unaffected by the multiplex transmission.

Diaphony and convolution

If we compute the functions $D_{e^i}(n)$ in the universal diaphony matrix, we state that this matrix is evidently symmetrical, but moreover, some terms from different Walsh functions pairs are found equal. The convolution theorem states that the terms in the sum (7) are the Fourier coefficients of the convolution product :

$$Q_{ij}(x) = \int_0^\pi W(i,u) \cdot W(j,u-x) du \quad (9)$$

Next section will be devoted to compute how many distinct convolution products exist, to within sign, in an equivalence class containing 2^k Walsh function.

Polynomial representation

In the sequel, we change sign normalization of Walsh functions so that :

$$W(N, -\pi + 0) = +1 \text{ for all } N.$$

We represent Walsh carriers in the ring of polynomials in y , $\text{mod}(y^{2^k}-1)$, with coefficients in the real numbers field without restriction.

In reference (1), we give in extenso the proofs of following theorems :

Theorem 2 - Any Walsh function can be factorized under the form :

$$P(y) = (1 \pm y)(1 \pm y^2)(1 \pm y^4) \dots (1 \pm y^{2^{k-1}}) \quad (10)$$

If we associate "+" to 0 and "-" to "1", the choice of signs in these k factors gives the rank N of the function in Gray code.

(This theorem is established by proving that 2^k orthogonal functions are so generated, and that if the number of zero-crossing of the function on the open definition interval increases by one, one sign only can change in (10)).

Theorem 3 - If two Walsh functions are represented by the polynomials :

$$P(y) = W(i, x) \\ P'(y) = W(j, x)$$

then the product : $Q(y) = P(y) \cdot P'(1/y) \text{ mod}(y^{2^k}-1)$ has for coefficients the successive values of the convolution product (9) at the beginning of each interval where Walsh functions are constant.

$$\text{If } Q(y) = c_0 + c_1 y + c_2 y^2 + \dots + c_{2^k-1} y^{2^k-1}$$

then :

$$Q_{ij}(m \pi / 2^{k-1}) = c_m \quad (m \text{ integer from } 0 \text{ to } 2^k-1)$$

(This theorem is easily proved by considering the triangular elementary convolution product of two functions equal to +1 on a single interval and to zero elsewhere, represented by single powers of y , and using then the bilinear character of the integral (9) for combining these building blocks into Walsh functions).

The convolution product $Q_{ij}(x)$ being a succession of straight segments, it is completely defined by its values just obtained on interval separative points.

We now prove the :

Theorem 4 - The convolution product of any two Walsh functions belonging to the same diaphonic equivalence class is uniquely defined, to within sign, by the C-sum : $e \pm e'$ of their equivalence numbers written in Gray code.

As a corollary :

In a diaphonic equivalence class containing 2^k functions, the C-sum is a k-digit ternary number, so that 3^k distinct convolution products exist.

Proof If we factorize both polynomials $P(y)$ and $P'(1/y)$ under the form (10) and effect the multiplication of corresponding parentheses, we find that, to within sign, the product of parentheses with degree 2^p can take only 3 forms and not 4, before the reduction $\text{mod}(y^{2^k}-1)$ is effected :

$$(1 + y^{2^p})(1 + 1/y^{2^p}) = 2 + y^{2^p} + 1/y^{2^p} \quad (\text{"-1" form})$$

$$(1 + y^{2^p})(1 - 1/y^{2^p}) = -(1 - y^{2^p})(1 + 1/y^{2^p}) = -2 - 1/y^{2^p}$$

($"0"$ form)

$$(1 - y^{2^p})(1 - 1/y^{2^p}) = 2 - y^{2^p} - 1/y^{2^p} \quad (\text{"+1" form})$$

As a consequence from theorem 2, the equivalence number in Gray code in a equivalence class defines the signs in the expression (10) for the functions in that class and

announced result follows.

Convolution classes

If two Walsh functions are in diaphonic equivalence classes with different values of k_0 , the convolution product vanishes identically: $Q_{ij}(x) = 0$ for all x .

If two Walsh functions are in diaphonic equivalence classes with the same index k_0 , their convolution product does not vanish, even if the functions have not the same parity and are absolutely orthogonal for this reason. (In all cases: $Q_{ij}(0) = 0$ if $i \neq j$).

We see that having a non-zero convolution product is an equivalence relation for Walsh functions. Equivalence classes for this relation are the union of two classes with same k_0 and opposite parities.

Finally, then: there are $k+1$ convolution classes in the group of the 2^k first Walsh functions.

All these classes are isomorphic as a consequence of Theorem 1.

Having in mind that any even Walsh function is equal to the preceding odd function cyclically shifted, we can recapitulate following results:

- in a diaphonic equivalence class containing 2^k Walsh functions (e from 0 to 2^{k-1}), there are, to within sign, 3^k different convolution products (Th. 4) and, then, 3^k distinct diaphonic characteristics $D_{ij}(n_{\max})$
- in a convolution class containing 2^{k+1} Walsh functions, and resulting from the union of two diaphonic classes with same k_0 , there are: $4 \cdot 3^k$ different convolution products to within sign, but only 3^k different convolution products to within a cyclical shift (Each product is repeated 4 times with different cyclical shifts).

Synchronization Error effect

Suppose now that in the multiplex fig. 1, there is neither line filtering nor additive noise, but that a time lag r exists between the clocks generating the Walsh carriers in transmission and reception ends. Let 1 be the period.

We suppose also that:

- 1) r is small with regard to an elementary

interval: $r \ll T/2^k$ (this means that synchronization is acceptable)

- 2) relative variation of the s_i over one period can be neglected (this approximation is tantamount to evaluating the main part only in the error).

Denote by $W'(N, t)$ the Walsh functions at receiving end:

$$W'(N, t) = W(N, t-r).$$

Received signals will be:

$$s'_{ij} = \frac{1}{T} \sum_{i=0}^{2^k-1} s_i \cdot \int_0^T W(i, t) \cdot W'(j, t) dt =$$

$$\frac{1}{T} \sum_{i=0}^{2^k-1} \int_0^T W(i, t) W(j, t-r) dt$$

$$\text{or: } s'_{ij} = \sum_{i=0}^{2^k-1} P_{ij}(r)$$

if $P_{ij}(r)$ is the normalized convolution product:

$$P_{ij}(r) = \frac{1}{T} \int_0^T W(i, t) \cdot W(j, t-r) dt \quad (P_{ij}(0) = \delta_{ij})$$

As a result of hypothesis 1, we have:

$$P_{ij}(r) = r \frac{dP_{ij}(0)}{dr} \quad \text{if } i \neq j, \text{ and } P_{ii}(r) = 1 - r \frac{dP_{ii}(0)}{dr}$$

and we study hereafter the absolute values K_{ij} of the slopes at origin of convolution products. These coefficients K_{ij} are the influence coefficients of synchronization error, which will produce on channel nr.i:

- an attenuation: $P_{ii}(r) = 1 - r |K_{ii}|$
- a diaphony from j th. channel: $P_{ij}(r) = r |K_{ij}|$

with $r > 0$.

Slopes at origin of convolution products

The complete (and rather tedious) proofs of following facts are exposed in extenso in ref. 2:

Theorem 5 - Slopes at origin K_{ij} are integer multiples of the smallest one. Inside a convolution class, they are odd integer multiples of it.

(This is a consequence of the c_j in theorem 3 being integers).

Theorem 6 - Slope at origin K_{ii} of an auto-convolution product is proportional to the sequency of the function.

$$K_{ii} = 4 s/T$$

(Each zero crossing in the function has a contribution $2 s/T$ in the decrease of the normalized autoconvolution product).

Theorem 7 - We have : $K_{ij} \leq \inf (K_{ii}, K_{jj})$

Universal Matrix for Synchronization error coefficients

A consequence of preceeding results is that we can construct an universal symmetrical matrix with integer terms : $k_{ij} = K_{ij} \cdot T/4$ for the normalized influence coefficients of synchronization error. All diagonal terms will be equal to the sequency of the corresponding Walsh function.

If we restrict this matrix to the first convolution class, $k_0 = 1$ it will comprise only odd integers. The k_i matrix for a class with $k_0 \neq 1$ will be obtained by multiplying it by 2^{k_0-1} (by theorem 1).

Following recursive construction gives the k matrix of rank 2^{p+1} , let it be \bar{k}_{p+1} , when the k matrix of rank 2^p is known (cf. 2 for complete justification).

Let \bar{k}_{p+1} be divided into 4 squares (fig. 3). Upper left square is \bar{k}_p . Upper right square is $S\bar{k}_p$ obtained from \bar{k}_p by column reversal. Lower left square is $S^T\bar{k}_p$, obtained from \bar{k}_p by line reversal. Lower right square is ?

$$\bar{k}_p + 2^p \cdot \bar{1} \quad (\bar{1} \text{ being the unitary matrix with rank } 2^p).$$

Starting matrix \bar{k}_1 is easily seen to be : $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ for the two functions of unit sequency.

$$\bar{k}_{p+1} = \begin{bmatrix} \bar{k}_p & S\bar{k}_p \\ S^T\bar{k}_p & \bar{k}_p + 2^p \cdot \bar{1} \end{bmatrix}$$

Fig. 3 Recursive construction of matrix \bar{k}_{p+1}

Figure 4 gives the \bar{k}_7 matrix, allowing to compute the synchronization effect for the 16 first Walsh Functions group.

$i \setminus j$	N	1	2	5	6	9	10	13	14
1	1	1	1	1	1	1	1	1	1
3	2	1	1	1	1	1	1	1	1
5	5	1	1	3	1	1	3	1	1
7	6	1	1	1	3	3	1	1	1
9	9	1	1	1	3	5	1	1	1
10	10	1	1	3	1	1	5	1	1
13	13	1	1	1	1	1	1	7	1
14	14	1	1	1	1	1	1	1	7

Fig. 4 The matrix \bar{k}_7

Conclusions

From an algebraic study of Walsh functions we have derived practical formulas to compute the effects of limited passband, noise and synchronization errors in a Walsh carrier multiplex. The consideration of equivalence classes leads to the conclusions that from the group of the 2^k first Walsh functions we can extract $2k$ carriers such that a multiplex with these carriers shall have no interchannel diaphony induced by the limited passband of the line, and $k+1$ carriers such that a multiplex with these carriers shall have no interchannel diaphony induced by the synchronization error.

These results are of practical meaning.

Our initial non recursive generation algorithm has analogies with the definition proposed by LACKEY and MELTZER (3) and justified in extenso by DAVIES (4). But the choice, as a basis, of even (cyclically shifted) Rademacher functions instead of odd Rademacher functions opens the way, by theorem 1, to a synthetic view of the spectral properties of Walsh functions.

References

- (1) CARDOT, C. Définition analytique simple des fonctions de Walsh et application à la détermination exacte de leurs propriétés spectrales.
Ann. Télécommunic., Fr, (1972),
27, n° 1-2, pp. 31-48
- (2) CARDOT, C. Etude de l'influence de l'erreur de synchronisation sur le fonctionnement d'un multiplex à porteuses Walsh.
Ann. Télécommunic., Fr, (1972),
27, n° 3-4.
- (3) LACKEY, R.B. and MELTZER, D.
A simplified Definition of Walsh Functions.
I.E.E.E. Trans. on Computers,
Vol. C-20, nr. 2 Feb. 71,
pp. 211 - 213.
- (4) DAVIES, A.C. On the Definition and Generation of Walsh Functions.
I.E.E.E. Trans. on Computers,
Vol. C-21, nr. 2 - Feb. 72.

SEQUENCY UNION

A Specialist Working Group of the IEEE Electromagnetic Compatibility Group

Since the beginning of the Walsh Functions Symposia in Washington, D.C. in 1969, there have been repeated suggestions that a scientific group should be organized for those working on applications of Walsh functions and other complete systems of nonsinusoidal functions. In September 1971 H. C. Andrews of the University of Southern California, H. F. Harmuth of Catholic University of America, G. S. Robinson of Comsat Laboratories and J. L. Walsh of the University of Maryland took the initiative and invited all those who had helped organize the symposia or had presented papers to form a Founding Committee for such a group. The following 47 scientists became members of this Founding Committee of the Sequency Union:

N. Ahmed (USA), H. C. Andrews (USA), R. Barrett (England), J. W. Bayless (USA), C. Boesswetter (Germany), E. Briganti (Italy), V. D. Brown (USA), E. C. Claire (USA), R. B. Crittenden (USA), I. Davidson (Canada), A. R. Elliott (Canada), T. H. Frank (USA), J. E. Gibbs (England), J. P. Golden (USA), J. A. Gordon (England), H. F. Harmuth (USA), H. A. Helm (USA), H. Huebner (Germany), T. Ito (Japan), P. C. Jain (USA), S. C. Kak (India), J. Kane (Canada), J. D. Lee (USA), R. Lopez de Zavalia (Argentina), P. V. Lopresti (USA), P. S. Moharir (India), G. G. Murray (USA), S. S. R. Murthy (India), J. Pearl (USA), K. R. Rao (USA), G. R. Redinbo (USA), G. S. Robinson (USA), J. H. Rosenbloom (USA), G. F. Sandy (USA), H. F. Schlicke (USA), P. E. Schmid (Switzerland), R. O. Schmidt (USA), H. H. Schreiber (USA), K. Shibata (Japan), G. J. Simmons (USA), W. Steenaart (Canada), L. W. Thomas (USA), H. Ueberall (USA), J. L. Walsh (USA), J. H. Wheelchel (USA), P. A. Wintz (USA), C. K. Yuen (Australia).

The objectives of the Sequency Union are to advance understanding in the uses of nonsinusoidal, complete, orthogonal sets of functions; to diffuse the knowledge of these uses; and to assist the members in these activities. This group should only exist until the use of nonsinusoidal functions is widely understood. If and when nonsinusoidal functions are used as routinely as sinusoidal

functions are used now, there would be as little need for a group promoting nonsinusoidal functions as there is now for a group promoting sinusoidal functions.

At the first meeting of the Sequency Union in December 1971, it was agreed to seek affiliation with the IEEE Electromagnetic Compatibility Group. This Group had supported the Walsh Functions Symposium in spring of 1971. Furthermore, the one problem common to all applications of nonsinusoidal functions in all fields of communications is that of compatibility with rules, methods and equipment intended for sinusoidal functions. Mr. L. W. Thomas negotiated affiliation with the EMC-Group. A formal invitation to join as a Specialist Working Group was extended by Mr. R. M. Showers, Chairman of the Technical Advisory Committee of the EMC-Group. This invitation was accepted at the meeting of the Founding Committee of the Sequency Union at Catholic University on 28 March 1972.

It is hoped that the IEEE EMC-Group will become the center for scientists interested in the use of complete, orthogonal systems of nonsinusoidal functions. The IEEE Transactions on Electromagnetic Compatibility are dedicated to problems of compatibility and interference. There are few uses of nonsinusoidal functions for which compatibility and interference are not problems, and this makes the IEEE Transactions on EMC a proper journal for the publication of papers on Walsh and other nonsinusoidal functions. General information on the activities of the Sequency Union will be published in the Newsletter of the EMC-Group, which is published quarterly and sent to all members of the Group.

Henning F. Harmuth
Chairman, Sequency Union
Department of Electrical Engineering
The Catholic University of America
Washington, D. C. 20017